МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Методика обучения решению задач на динамическое программирование для профильных классов

Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность программы бакалавриата
«Информатика. Экономика»

Проверка на объем заимствований:
67,78 % авторского текста

Работа рекомендована к защите
рекомендована/не рекомендована

«25» мая 2018 г.
и.о. зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Выполнил:
Студент группы ОФ-513/094-5-1
Терехов Антон Олегович

Научный руководитель:
Старший преподаватель
Боровская Елена Владимировна

Челябинск
2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Методика обучения решению задач на динамическое программирование для профильных классов

Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями подготовки)
Направленность программы бакалавриата
«Информатика. Экономика»

Проверка на объем заимствований:
_____% авторского текста

Работа _____ к защите
рекомендована/не рекомендована

« ___ » _____ 20__ г.
и.о. зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Выполнил:
Студент группы ОФ-513/094-5-1
Терехов Антон Олегович

Научный руководитель:
Старший преподаватель
Боровская Елена Владимировна

Челябинск
2018

Оглавление

ВВЕДЕНИЕ 3
Глава 1. Теоретические основы решения задач методом динамического программирования 6
1.1 Основное понятия метода динамического программирования 6
1.2 Виды классических задач, решаемых методом «Динамическое программирование» 12
1.3 Динамическое программирование в ЕГЭ 19
1.4 Олимпиадные задачи решаемые, методом динамического программирования 25
1.5 Анализ существующих обучающих материалов по динамическому программированию 30
Выводы по I главе 34
Глава 2. Разработка элективного курса «Динамическое программирование» 35
2.1 Анализ нормативных документов 35
2.2 Элективный курс «Динамическое программирование» 39
2.3 Программно-методическая поддержка элективного курса 60
2.4 Апробация результатов исследования в школе 63
Выводы по II главе 64
ЗАКЛЮЧЕНИЕ 65
Библиографический список 66

ВВЕДЕНИЕ

В современном обществе объемы информации превышают весь информационный объем планеты, существовавший 20 лет назад. Процессы обработки всей поступающей информации занимают много времени. Они развиваются, становятся все более и более сложными с точки зрения математики, программирования. Процесс быстрой и качественной обработки информации – является сложной задачей. Разбивая сложные задачи на множество простых – мы снижаем затраты времени на вычисления и отработку необходимых процессов для достижения максимального результата. Если для одношаговых процессов принимаемые решения, как правило, относительно просты, то в многошаговых процессах структура решения несравнимо сложнее. Решать эти задачи очень сложно не только для начинающего, но и для опытного программиста.

В 1940 году Ричардом Беллманом был впервые разработан метод, который является важнейшей основой для решения сложных современных задач. «Метод Беллмана» или «Метод динамического программирования» – широко известный математический метод для решения сложных задач путем разбиения их на более простые пересекающиеся подзадачи.

В 1953 году Р. Беллман применил метод динамического программирования к ряду конкретных практических задач, которые решались с помощью принципа оптимальности. Областью применения метода являлись многошаговые процессы, которые развивались во времени отличаясь от статического, линейного и математического программирования.

Вся накопившая теория, статьи, научные работы по динамическому программированию созданы лишь для “продвинутых” программистов получающих профессиональное образование [18]. В то же время, метод динамического программирования используется для решения олимпиадных задач по информатике и заданий ЕГЭ. Однако, отсутствуют методические разработки обучения динамическому программированию школьников. На

данный момент в общеобразовательной школе (в том числе и в профильных классах) выделяется недостаточно времени на изучение динамического программирования.

Таким образом, разработка методики обучения динамическому программированию в школьном курсе информатики актуальна.

Целью квалификационной работы является, разработка элективного курса «Динамическое программирование» для учащихся профильных классов.

Объект исследования – обучение учащихся информатике.

Предмет исследования - обучение учащихся профильных классов (по информатике) решению задач с использованием метода динамического программирования**.**

Гипотеза исследования: Если ввести для учащихся профильных классов элективный курс «Динамическое программирование», то это будет способствовать дальнейшему развитию алгоритмического мышления школьников, а так же более успешному решению олимпиадных задач по информатике и задач ЕГЭ, решаемых методом динамического программирования.

Для достижения поставленных целей были сформулированы следующие **задачи**:

1. Исследовать историю возникновения метода динамического программирования, основные понятия.
2. Проанализировать виды задач, решаемых методом динамического программирования.
3. Рассмотреть виды задач решаемых методом динамического программирования, представленные в ЕГЭ.
4. Рассмотреть примеры задач решаемых методом динамического программирования, представленные на российских олимпиадах.
5. Проанализировать существующие учебные пособия по динамическому программированию для школьников.

6. Разработать элективный курс «Динамическое программирование» с программно-методической поддержкой в виде сайта.
7. Провести апробацию элективного курса в школе.

Глава 1. Теоретические основы решения задач методом динамического программирования

1.1 Основное понятия метода динамического программирования

В истории первым понятие динамического программирования ввел Ричард Беллман в 1940 году. В то время ученый работал в качестве научного сотрудника корпорации RAND. Он был профессором Университета Южной Калифорнии. Беллман удостоился Медали почета от Института инженеров электротехники и электроники (IEEE) в 1979-м, за вклад в теорию процессов принятия решений и теорию управления системами, в частности, за создание и применение динамического программирования, опубликовал 619 статей и 39 книг[10].

Ключевой работой Р. Беллмана стало «уравнение Беллмана» или «уравнение динамического программирования». Впервые уравнение Беллмана было применено к теории управления, междисциплинарной области инженерии и математики, а также к другим темам в области прикладной математики. Впоследствии уравнение Беллмана стало важным инструментом в экономической теории.

Это уравнение является достаточным условием для оптимальности, ассоциируемой с математическим методом оптимизации, известным как динамическое программирование. Р. Беллман в период своей научной деятельности получил многочисленные результаты, связанные с применением динамического программирования в разных областях математики (вариационное исчисление, автоматическое регулирование, теория аппроксимации, исследование операций и др.). Почти к любой проблеме, которая может быть решена с помощью теории оптимального контроля, можно подобрать решение путем анализа уравнения Беллмана.

Понятие динамического программирования относилось к разделу прикладной математики «исследование операций».

Исследование операций – это построение, разработка и приложение математических моделей принятия оптимальных решений [11].

Оптимальное решение задачи – это аспект исследования операций, в который включается анализ и решение математических задач выбора в заданном множестве допустимых решений X элемента, удовлетворяющего определенным критериям оптимальности. Такие задачи, ищущие оптимальное решение, называются оптимизационными. Прикладной аспект исследования операций состоит в составлении и реализации оптимизационных задач[12].

Задачи исследования операций с любым множеством допустимых решений X и с весьма произвольными критериями оптимальности, могут иметь вид требований о максимизации (или минимизации) значений некоторой числовой или векторной функции на X. Эта функция обычно называется целевой функцией. В первом случае говорят о задаче математического программирования (оптимального программирования, что не следует смешивать с программированием на компьютере), а во втором задаче векторной оптимизации, или о многокритериальной задаче.

Математическое программирование - математическая дисциплина, посвященная теории и методам решения задач о нахождении экстремумов функций на множествах в конечномерных векторных пространствах, определяемых линейными и нелинейными ограничениями (равенствами и неравенствами). Математическое программирование — раздел науки об исследовании операций, охватывающий широкий класс задач управления, математическими моделями которых являются конечномерные экстремальные задачи. Наименование «математическое программирование» связано с тем, что целью решения задач является выбор программы действий.

В математическом программировании принято выделять следующие разделы:

Линейное программирование — целевая функция и ограничения линейны; квадратичное программирование — целевая функция квадратична

и выпукла, допустимое множество определяется линейными равенствами и неравенствами;

Выпуклое программирование — целевая функция и допустимое множество выпуклы;

Дискретное программирование решение ищется лишь в дискретных, например целочисленных, точках допустимого множества значений переменных;

Стохастическое программирование в отличие от детерминированных задач здесь входная информация носит элемент неопределенности.

Круг задач решаемых методом динамического программирования был четко обозначен.

Метод на тот момент включался в одну линейку с методами оптимальных(наилучших) решений в задачах управления системами с одним отклонением. Если методы линейного и математического программирования были статичными, то в динамическом программировании система от которой искалось решение могла меняться в течении времени.

Особенность метода по сравнению с другими заключается в том, что весь процесс управления системой и саму систему решения задачи можно было разбить на фиксированные промежутки времени, где оптимальное решение задачи складывается из оптимальных решений каждого промежутка (этапа, шага). Поэтому метод был назван «динамическим» – изменяющимся во времени

В то время термин «динамическое программирование» (dynamic program) никак не был связан с современным, который учитывает, что динамическое программирование используется в создании программного – компьютерного решения в виде кода, который вычисляется на устройстве ЭВМ.

Динамическое программирование в математике и теории вычислительных систем (dynamic program) — метод решения задач с оптимальной подструктурой и перекрывающимися подзадачами, который

намного эффективнее, чем полный перебор[7].

В ходе дальнейшего развития 1953 году – динамическое программирование уже не вкладывалось как раздел прикладной математики. Оно стало относится к информатики в целом и обрело свое название.

Динамическое программирование (dynamic program) – это способ решения сложных задач путем разбиения их на более простые подзадачи, где на каждый шаг решения задачи выделяется фиксированное время. Задача, к которой применяется метод динамического программирования должна содержать оптимальную подструктуру выглядящей, как набор перекрывающихся подзадач, сложность которой меньше исходной[9].

Оптимальная подструктура в динамическом программировании означает, что оптимальное решение подзадач меньшего размера может быть использовано для решения исходной задачи[14].

Перекрывающиеся подзадачи в динамическом программировании означают подзадачи, которые используются для решения некоторого количества задач (не одной) большего размера (то есть мы несколько раз проделываем одно и то же)[14].

Динамическое программирование помимо задач оптимизации теперь очерчивало вполне определенный метод проектирования алгоритмов. Конечно метод не универсален и не всегда подходит к заданным условиям задачи. Задачи, к которым теперь применялся метод динамического программирования уже давно известен и рассматривается в пункте 1.2 квалификационной работы.

Разбивка задачи на подзадачи это стандартная практика построения любого алгоритма – универсальный метод, которым пользуются все программисты. Но суть динамического программирования заложен в двух терминах: «разбивка», «компоновка». Решаемая задача должна допускать разбивку на подзадачи того же вида так, чтобы решение как бы складывалось, компоновалось из оптимальных решений подзадач. То есть должны быть определены рекуррентные соотношения – зависимости между

подзадачами. Особенностью метода является то, что каждую подзадачу мы решаем только 1 раз, запоминая ее решение. Результат ее оптимального решения запоминается на каждом шаге и затем при решении последующих задач используется как данное. Получаются два основных момента при переходе от одной подзадачи к другой: «связь» и «запоминание».

Метод динамического программирования выглядит со стороны простым и как возможно показывается может быть применим к решению любой задачи, но не все так просто. Универсальных рекомендаций о применении метода к задачи не существует! Никто еще не разработал динамическую схему к каждой задачи. Это необходимо понимать, ведь бывает, что подобрать решение методом динамического программирования в определенны случаях довольно сложно. Кроме этого отсутствуют строгие правила леммы, теоремы и т. Д. это не случай какой-то теоремы, познавать метод можно только от практики решения конкретных задач. Именно поэтому вся теория динамического программирования содержит в себе 90% практики и только 10% теории.

Исследуя понятие и метод динамического программирования можно выделить 4 принципа, которые его образуют и используются при решении задач.

1. Сложная задача всегда разбивается на множество подзадач до того момента, пока решаемая задача не станет тривиальной. Проще говоря это сведение основной задачи к более простым того же типа.

2. ДП решает задачи, которые требуют полного перебора всех вариантов решения:

2.1 «подсчитайте количество вариантов…»

2.2 «как оптимально распределить…»

2.3 «найдите оптимальный маршрут…»

3. Принцип оптимальности проявляется в том, что оптимальное решение принимаемое на каждом шаге решения задачи, не зависит от предыстории. Оптимальное решение выбирается с учетом факторов,

характеризующих процесс в данный момент. Примером может служить выбор кратчайшего пути в графе из одной контрольной точки в другую, промежуточного пункта в конечный, мы принимаем решение независимо от того, как, когда и какой дорогой он прибыл в данный пункт. Мы руководствуемся лишь расположением пункта в сети дорог.

4. При решение задач ДП требуется дополнительная память для ускорения решения задачи, ведь на каждом шаге мы запоминаем решение всех подзадач с наименьшим или наибольшим значением параметров.

В динамическом программировании существует два подхода к решению задач:

1. Нисходящее динамическое программирование: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. Используется запоминание для решений часто встречающихся подзадач.

2. Восходящее динамическое программирование: все подзадачи, которые впоследствии понадобятся для решения исходной задачи просчитываются заранее и затем используются для построения решения исходной задачи. Этот способ лучше нисходящего программирования в смысле размера необходимого стека и количества вызова функций, но иногда бывает нелегко заранее выяснить, решение каких подзадач нам потребуется в дальнейшем.

Все виды задач, решаемые методом динамического программирования, решаются по следующему алгоритму:

1) выделить и описать подзадачи, через решение которых будет выражаться искомое решение, строение оптимальных решений;

2) выписать рекуррентные соотношения (уравнения), связывающие оптимальные значения параметра для подзадач;

3) вычислить оптимальное значение параметра для всех подзадач;

4) построить оптимальное решение, используя полученную информацию.

Если в отличие от поиска оптимального пути нам встречается задача подсчета количества допустимых вариантов или решений по определенному условию, то шаг 4 в алгоритме будет отсутствовать.

Однако в 3 шаге, приходится совершать дополнительное действие по сохранению всех ранее выполненных решений алгоритма в виде дополнительной информации.

При решении задач обычно самым сложным оказывается реализация именно 4 шага.

Как можно заметить, данная теория сложна для ученика, что показывает необходимость адаптации материала для элективного курса.

1.2 Виды классических задач, решаемых методом «Динамическое программирование»

В теории динамического программирования с 1940 года было рассмотрено множество задач, конкретных примеров с определенными условиями затрагивающих разные области как математики, так и программирования. Р. Беллман внес свой вклад рассмотрев часть задач, подробно расписав несколько задач показывающих оптимальное решение и применение уравнения Беллмана. Метод динамического программирования расширялся, прорабатывался другими учеными, которые рассматривали задачи из своей практической деятельности создавая новые виды и дополняя набор классических задач.

К видам задач решаемых методом динамического программирования относятся те, которые из-за условия имеют схожее рекуррентное соотношение, одинаковое уравнение Беллмана. Со времени создания динамического программирования, как отдельного метода, требующего специальные знания, прошло много времени, как и в науке математике, в нем сформировалась группа классических задач. Конечно же встречаются и другие задачи в практической деятельности программиста, но данный классификатор отражает основные виды, которые необходимо знать, чтобы специалист разбирался в самом методе.

Классические задачи метода динамического программирования:

1. Задача о вычислении чисел Фибоначчи;
2. Задача о наибольшей общей подпоследовательности;
3. Задача поиска наибольшей увеличивающейся подпоследовательности;
4. Задача о редакционном расстоянии (расстояние Левенштейна);
5. Задача о порядке перемножения матриц;
6. Задача о выборе траектории;
7. Задача последовательного принятия решения;
8. Задача об использовании рабочей силы;
9. Задача управления запасами;
10. Задача о ранце;
11. Алгоритм Флойда — Уоршелла: найти кратчайшие расстояния между всеми вершинами взвешенного ориентированного графа;
12. Алгоритм Беллмана — Форда: найти кратчайший путь во взвешенном графе между двумя заданными вершинами;
13. Максимальное независимое множество вершин в дереве.

Каждая задача может быть решена с помощью рекурсии или полного перебора всех вариантов, где вычисления решения займут не один день и вычислительная мощность должна быть равна суперкомпьютеру. Выполнение большого числа операций требует большого объема времени, что невыгодно при решении.

Однако среди таких задач можно выделить типы, которые могут быть решены за счет особого свойства – имея решения некоторых подзадач, можно найти решение исходной задачи.

Именно такие задачи решаются методом динамического программирования. Рассмотрим их последовательно.

1. Задача о вычислении чисел Фибоначчи

Сформулируем задачу: Найти по заданному числу n член последовательности Фибоначчи F_n.

Последовательность Фибоначчи – последовательность в которой каждый последующий член прогрессии равен сумме двух предыдущих - 1, 1, 2, 3, 5, 8, ...(Табл. 1)

Таблица 1

Данные задачи Фибоначчи

Формат входных данных	**Формат выходных данных**
6	8

Рассмотрим задачу с точки зрения решения методом динамического программирования[1].

Чтобы найти последовательность, нам нужно как-то обозначить каждый последующий член и записать формулу рекуррентного соотношения, чтобы можно было вычислить каждый последующий член.

Пусть Fn – n–ный ($n \geq 0$) член последовательности Фибоначчи. $F_0 = 0$ – Нулевой член последовательности, $F_1 = 1$ – Первый член последовательности, $F_2 = 1$ – Второй член последовательности, $F_3 = F_2+F_1$ – Третий член последовательности, и т.д.

Тогда формула рекуррентного соотношения:

$F_n=F_{n-1}+F_{n-2}$, где $n>1$, $F_0 = 0$, $F_1 = 1$.

Прежде чем переходить к решению задачи, стоит ограничить диапазон чисел укладывающихся в решение задачи типом переменной n – LongInt, для больших значений требуются специальные вычисления.

```
Function Fs(n: LongInt): LongInt;
Begin
        If (n = 1) or (n = 2) then Fs:= 1
         Else Fs:= Fs(n - 1) + Fs(n - 2);
End;
```

Логика решений будет иметь следующий вид Рис 1. Задача будет иметь экспоненциальную временную сложность $O(2^n)$.

Решим задачу методом динамического программирования.

Первым действием нам необходимо определить понятие подзадачи. В данном случае – это вычисление нового числа последовательности Фибоначчи. Для каждого i-го члена, мы решаем очередную задачу $F_n=F_{n-1}+F_{n-2}$ и так следуя к необходимому указанному в задаче позиции искомого числа n(Рис. 1).

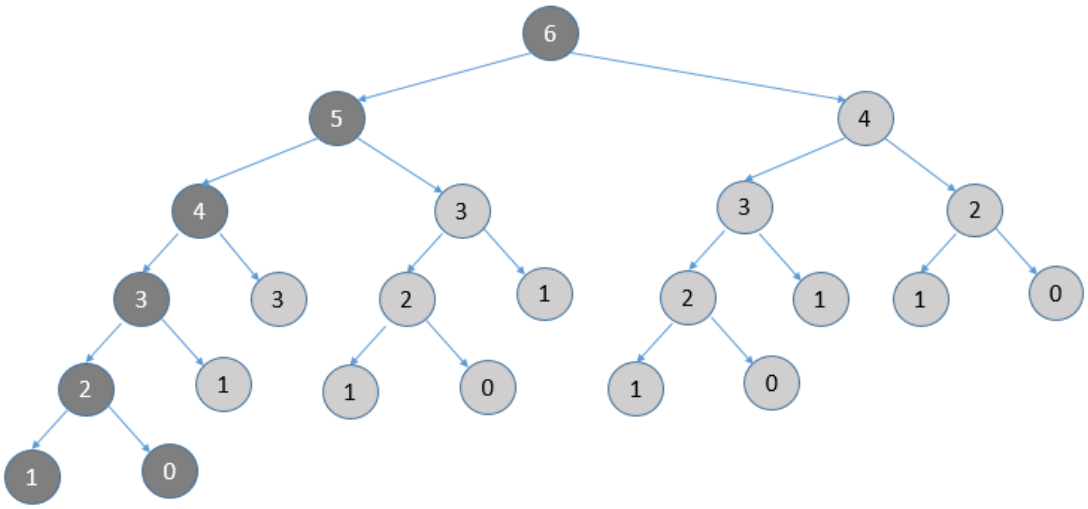

Рисунок 1. Дерево решения

При этом решая каждую подзадачу вычисления нового числа только один раз. Рассмотрев рисунок 1 и рекурсивное решение задачи, мы придём к выводу, что при решении задачи мы проходим только по темно-серым кружкам.

Предположим, что все результаты решения задачи мы записывали в массив F(Табл. 2)

Таблица 2

Массив данных

i	0	1	2	3	4	5	6
F[i]	0	1	1	2	3	5	8

Тогда решения имеет следующий вид:

```
Procedure Fs2(n: LongInt);
Var i: LongInt;
```

```
Begin
F[0]:=0;
F[1]:=1;
F[2]:=1;
    For i:=3 to n do
    F[i]:= F[i-1] + F[i-2];
Writeln(F[n]);
End;
```

Данный пример решения задачи не учитывает то, что для решения следующей подзадачи нам необходимы только два решения предыдущих подзадач. Значит массив F не подходит для задачи. Нам требуется лишь 2 переменных сохраняющих предыдущие значения.

Преобразуем решение:

```
Procedure Fs3(n: LongInt);
Var i,a,b,c: LongInt;
Begin
a:=0;
b:=1;
    For i:=2 to n do
    Begin
    c:=a+b;
    a:=b;
    b:=c;
    end;
Writeln(c);
End;
```

Временная сложность вычислений – O(n).

Решив данную задачу, можно понять, как используется метод динамического программирования. Как выделяется подзадача, находится рекуррентная формула, запоминаются решения и все подзадачи стремятся к

исходному заданию. Каждую подзадачу мы решали только один раз, запоминая результат в массив F или в две переменные.

В результате нашего решения и сохранения данных подзадач, мы получили выигрыш во времени по сравнению с начальным рекурсивным вариантом задачи, где время экспоненциально велико. Время решения задачи трансформировалось в более эффективное решение, быстрее чем линейное.

2. ***Задача о наибольшей общей подпоследовательности***

   Даны две последовательности целых чисел.

   Найти общую подпоследовательность максимальной длины из указанных последовательностей(Табл. 3).

Таблица 3

Данные задачи НОП

Формат входных данных	**Формат выходных данных**
2,3,5,4 1,2,3,4	2,3,4

Пусть общая последовательность будет обозначена - Z. Тогда первая подпоследовательность содержащаяся в Z запишем как X, вторую подпоследовательность как Y. Элементы внутри последовательностей, каждая буква буду обозначаться так: Z = {z1, z2, …, zk} одна из НОП для X = {x1, x2, …, xn} и Y = {y1, y2, …, ym}(Табл. 4). В данном случае k,m,n – длины последовательностей.

Обозначим длину наибольшей общей последовательности (далее НОП) L.

Предположим, что длина одной из подпоследовательностей например n=0, тогда $L_{0n}=0$. Обратное верно и для длины последовательности Y. При m=0 НОП $L_{m0}=0$.

Мы будем искать решение НОП для всех i-первой последовательности, j-второй последовательности, то есть L_{ij}(табл. 4). (i >0, j >0)

Таблица 4

Массив данных X и Y

X	x1	**x2**	x3	xn
	2	3	5	4
Y	y1	**y2**	y3	ym
	1	2	3	4

Подзадачей для метода динамического программирования будет являться выбор сочетания цифр для НОП по рекуррентной формуле[4].

Решая данную задачу перебором мы можем столкнуться с двумя ситуациями: когда $x_i=y_j$ или $x_i \neq y_j$.

В первом случае нам становится ясно, что элементы x_i и y_j входят в НОП. Элементы x_i и y_j мы можем исключить из решения, т.к. они уже входят в НОП. Получается что нам остается решить $L_{(i-1)(j-1)}$. Стоит также не забывать, что мы уже получили один из вариантов НОП, а значит после решения задачи $L_{(i-1)(j-1)}$, нужно добавить 1 в L.

Во втором случае, когда $x_i \neq y_j$. Пусть x_i попадет в НОП. Тогда в НОП он будет занимать только последнее место. Но рассуждая также можно и y_j добавить в НОП. Он тоже попадет на последнее место в последовательности. Но $x_i \neq y_j$ –разные, то обязательно x_i или y_j не попадет в НОП. Учитывая такой вариант, что x_i – не попадет в НОП, нам останется решить задачу $L_{(i-1)j}$. Если не попадает в НОП y_j, тогда мы будем решать задачу $L_{i(j-1)}$.

По заданию нам требуется найти наибольшую последовательность, а значит мы возьмем наибольший $\max(L_{(i-1)j}, L_{i(j-1)})$.

Таким образом мы создали рекуррентное соотношение - уравнение Беллмана:

$$L_{i,j} = \begin{cases} L_{(i-1)(j-1)} + 1, \text{при } x_i = y_j \\ \max\left(L_{(i-1)j}, L_{i(j-1)}\right), \text{при } x_i \neq y_i \end{cases}$$

Временная сложность данной задачи будет O(n*m).

Решение задачи методом динамического программирования на языке Паскаль:

```
var x,y,z:string; a:array[0..250,0..250] of byte; i,j:byte;
begin
readln(x); readln(y);
fillchar(a,sizeof(a),0);
for i:=1 to length(x) do
for j:=1 to length(y) do
if x[i]=y[j] then a[i,j]:=a[i-1,j-1]+1
else if a[i-1,j]>=a[i,j-1] then a[i,j]:=a[i-1,j]
else a[i,j]:=a[i,j-1];{длина НОП найдена в a[length(x),length(y)]}
z:='';{строим саму НОП}
i:=length(x);
j:=length(y);
while (i>0) and (j>0) do
if x[i]=y[j] then begin z:=x[i]+z; i:=i-1; j:=j-1 end
else if a[i-1,j]>=a[i,j-1] then i:=i-1 else j:=j-1;
writeln(z);
end.
```

1.3 Динамическое программирование в ЕГЭ

В ЕГЭ существует набор документов кодификатор и спецификатор, которые отражают что необходимо знать учащемуся, чтобы он смог сдать экзамен и продолжить обучение в высшем учебном заведении.

Изучение метода динамического программирования неотъемлемая часть обучения профессионального программиста. В ЕГЭ представлены задачи решаемые методом динамического программирования. Если рассматривать современные задания ЕГЭ, то можно заметить тенденцию встречи данных задач под номерами 15, 22. В данной курсовой работе будут рассмотрены последние примеры данных задач из ЕГЭ 2018 года.

Рассмотрим спецификатор, включающий перечень требований к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике и ИКТ[6]. Выделим

умения необходимы для решения задач методом динамического программирования отразив их в таблице(Табл. 5)[5].

Таблица 5

Проверяемые элементы заданий КИМ ЕГЭ

№	Проверяемые элементы содержания	Коды проверяемых элементов содержания (по коду кодификатора)	Коды проверяемых требований к уровню подготовки (по кодификатору)	Уровень сложности задания
15	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	1.3.1 Оценивать объем памяти, необходимый для хранения информации	1.2.1 Использовать готовые модели, оценивать их соответствие реальному объекту и целям моделирования	П
22	Умение анализировать результат исполнения алгоритма	1.6.2 Вычислимость. Эквивалентность алгоритмических моделей	1.1.3 Строить информационные модели объектов, систем и процессов в виде алгоритмов	П

Уровни сложности заданий: Б – базовый; П – повышенный; В – высокий.

Кодификатор и спецификатор очень точно отображают необходимый объем знаний, который школьник должен освоить. Но к сожалению, в кодификаторе и спецификаторе полностью отсутствует упоминание, что задача может быть решена методом динамического программирования.

Рассмотрим задания 15,22 представленные в ЕГЭ и решим их методом динамического программирования.

Задание ЕГЭ № 15. Графы.

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город Л(Рис. 2)?

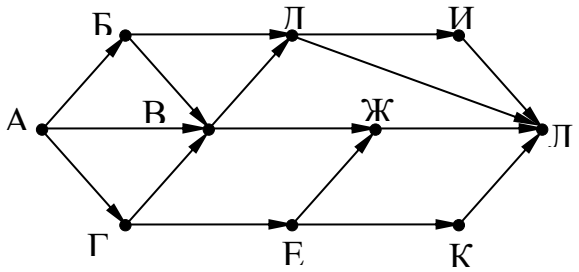

Рисунок 2. Схема дорог

Решение:

1) Будем обозначать через N_X количество различных путей из города А в город Х

2) Для города А есть только один маршрут – никуда не двигаться, поэтому $N_A = 1$

3) Для любого города Х количество маршрутов N_X можно вычислить как

$N_x = N_y + \ldots + N_z$,

где сумма взята по всем вершинам, из которых есть прямой путь в вершину Х; например,

$N_Л = N_И + N_Ж + N_К$

4) Около каждого города будем записывать количество маршрутов из А в этот город

5) Начнем считать количество путей с начала маршрута – с города А(Рис. 3):

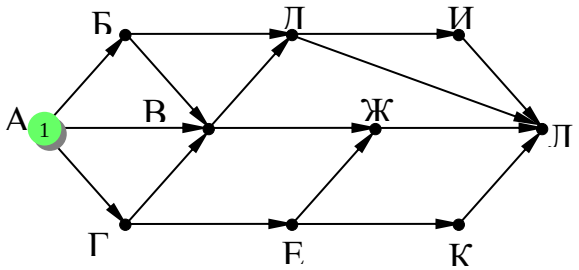

Рисунок 3. Схема дорог шаг 1

6) Теперь находим те вершины, в которые можно попасть напрямую из уже рассмотренных вершин (пока – только из А), это Б и Г, для них тоже количество путей равно 1(Рис. 4):

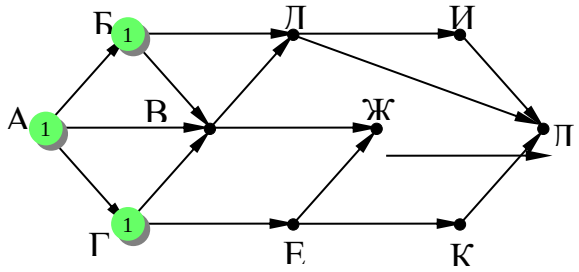

Рисунок 4. Схема дорог шаг 2

7) Теперь можно определить количество путей для В и Е; в В можно приехать только из А, Б и Г, а в Е – только из Г(Рис. 5):

$$N_В = N_А + N_Б + N_Г = 1 + 1 + 1 = 3$$

$$N_Е = N_Г = 1$$

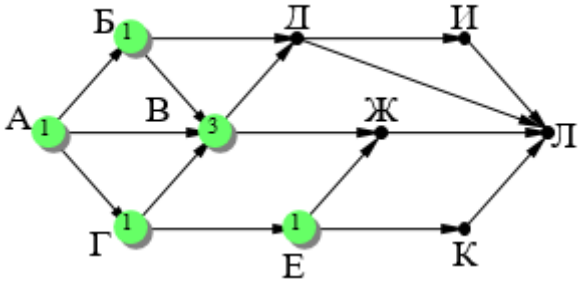

Рисунок 5. Схема дорог шаг 3

8) Теперь можно определить количество путей для Д, Ж и К; в Д можно приехать только из Б и В, в Ж – из В и Е, а в Е – только из Г(Рис. 6):

$$N_Д = N_Б + N_В = 1 + 3 = 4$$

$$N_Ж = N_В + N_Е = 3 + 1 = 4$$

$N_К = N_Е = 1$

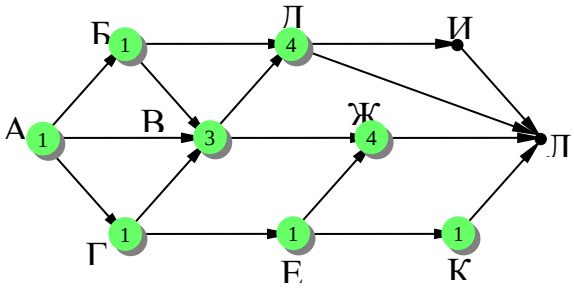

Рисунок 6. Схема дорог шаг 4

9) Теперь можно определить количество путей для И, куда можно приехать только из Д ($N_И = N_Д$) и, наконец, для Л(Рис. 7):

$N_Л = N_Д + N_И + N_Ж + N_К = 13$

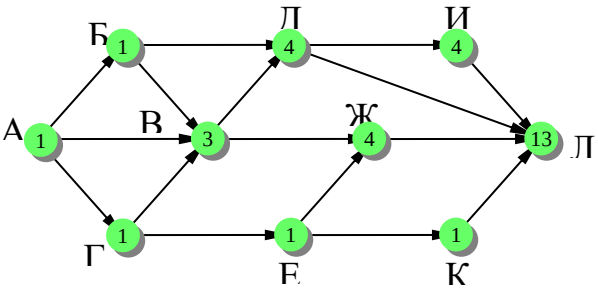

Рисунок 7. Схема дорог шаг 5

10) Ответ: 13.

Задание ЕГЭ № 22. Исполнитель калькулятор.

Исполнитель калькулятор преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1

2. Прибавить 2

3. Умножить на 3

Первая команда увеличивает число на экране на 1, вторая – увеличивает его на 2, а третья – умножает его на 2. Программа для исполнителя М17 – это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 12 и при этом траектория вычислений содержит числа 8 и 10?

Решение:

Запишем рекуррентную формулу для вычисления K_N – количества возможных программ для получения числа N из некоторого начального числа:

$K_N = K_{N-1} + K_{N-2}$, если N не делится на 3

$K_N = K_{N-1} + K_{N-2} + K_{N/3}$, если N делится на 3

Все допустимые программы можно разбить на 3 части(Рис.8):

– переход от 2 до 8

– переход от 8 до 10

– переход от 10 до 12

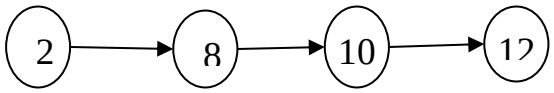

Рисунок 8. Программные части

Обозначим через $K_{a\to b}$ количеств возможных программ получения числа b из числа a

Очевидно, что $K_{a\to b} = K_{a\to c} \cdot K_{c\to b}$ для любого c, такого что $a < c < b$

Поэтому $K_{2\to 12} = K_{2\to 8} \cdot K_{8\to 10} \cdot K_{10\to 12}$

Вычисляем эти значения отдельно стандартным способом по рекуррентным формулам п. 1(Табл. 6.):

Таблица 6

Значения вычислений

N	2	3	4	5	6	7	8
K_N	1	1	2	3	6	9	15

N	8	9	10
K_N	1	1	2

10	11	12
1	1	2

и перемножаем: $15 \cdot 2 \cdot 2 = 60$

Ответ: 60.

1.4 Олимпиадные задачи решаемые, методом динамического программирования

В последнее время из-за роста интереса в мире программирования к вычислению, обработке большого объема данных в олимпиадных задачах увеличилось использование задач решаемых методом динамического программирования

Если раньше для решение задачи требовалось обработать единожды вводимую цепочку данных не более 255 символов, в современных задачах олимпиад требуется работа с средними объемами данных, которые раньше считались большими и могли обрабатываться несколько дней.

Развитие языков программирования, программного обеспечения и машинной части, повышение вычислений, увеличения банков данных требует от специалиста знания новых методов решения задач.

Метод динамического программирования придуманный Р. Беллманом в 1940 году использовался не слишком часто. Метод сложен в понимании и применении к задачам. Построить алгоритм, рекуррентное соотношение для задачи может не каждый специалист

Процесс глобализации с открытием информационных-экономических границ, увеличение населения способствовали созданию новых огромных баз данных

В качестве примера можно рассмотреть задачи решаемые методом динамического программирования представленные в олимпиадах 2015-2018 года. Ниже приводится описание задач с пояснениями.

Пример 1. Без двух нулей подряд

Требуется вычислить количество N-значных чисел в системе счисления с основанием K, таких что их запись не содержит двух подряд идущих нулей.

Входные данные:

Во входном файле INPUT.TXT записаны два натуральных числа N и K в десятичной системе счисления ($2 \le K \le 10$; $2 \le N$; $4 \le N+K \le 18$).

Выходные данные:

В выходной файл OUTPUT.TXT необходимо вывести целое число в десятичной записи – ответ на задачу(Табл. 7)[2].

Талица 7

Примеры входных и выходных данных «Без двух нулей подряд»

№	INPUT.TXT	OUTPUT.TXT
1	**2 10**	**90**
2	**4 2**	**5**
3	**6 3**	**328**

Для решения данной задачи используем метод динамического программирования. Пусть d[i] - количество чисел, состоящих ровно из i цифр и оканчивающихся на цифру, отличную от нуля. Аналогично, обозначим через $d_0[i]$ количество i-значных чисел, оканчивающихся на нуль. При этом будем рассматривать только те числа, которые не содержат в своей записи двух нулей подряд. Так же не будем рассматривать числа с лидирующими нулями, даже в том случае, когда i=1 (так как у нас n>1). Тогда очевидно, что d[1]=k-1 и $d_0[1]=0$. Теперь мы можем выразить d[i] и d0[i] через d[i-1] и d0[i-1] следующим образом:

$$d[i] = (d[i-1]+d_0[i-1])*(k-1),$$

$$d_0[i] = d[i-1].$$

Используя данные рекуррентные соотношения, мы можем линейно, пробегая по i от 2 до n, вычислить d[n] и $d_0[n]$. Ответом на задачу будет значение $d[n]+d_0[n]$. Вышеописанный алгоритм решения представим в следующей форме:

```
read(n,k)
d[1]=k-1; d0[1]=0
for i=2..n{
  d[i] = (d[i-1]+d0[i-1])*(k-1)
  d0[i] = d[i-1]
```

```
}
write(d[n]+d0[n])
```

Следует заметить, что ограничения на n и k в задаче позволяют использовать 4-байтовый целый тип для всех переменных. Использование массивов в данной задаче так же не обязательно, так как вычисляемые значения на i-м шаге зависят только от (i-1)-го шага.

Пример 2. Ход конем

Шахматная ассоциация решила оснастить всех своих сотрудников такими телефонными номерами, которые бы набирались на кнопочном телефоне ходом коня. Например, ходом коня набирается телефон 340-49-27. При этом телефонный номер не может начинаться с цифры 0, или с цифры 8[3].

Требуется написать программу, определяющую количество телефонных номеров длины N, набираемых ходом коня.

Входные данные:

Входной файл INPUT.TXT содержит натуральное число N ($N \leq 100$).

Выходные данные:

В выходной файл OUTPUT.TXT выведите искомое количество телефонных номеров(Табл. 8).

Талица 8

Примеры входных и выходных данных «Ход конем»

№	INPUT.TXT	OUTPUT.TXT
1	1	8
2	2	16

Для решения этой задачи применим метод динамического программирования. Пусть b[d][k] – количество номеров, набираемых ходом коня, которые начинаются с цифры d и состоят из k цифр. Тогда b[d][1]=1 для всех d, а b[d][k] для любого d вычисляется через сумму b[i][k-1] для k>1. Так, например, b[4][k] = b[0][k-1]+b[3][k-1]+b[9][k-1]. Увеличивая k от 2 до n

мы получим значения b[d][n], сумма которых (за вычетом b[0][n] и b[8][n]) и даст ответ на поставленную задачу.

Заметим также, что при вычислении могут получаться достаточно большие значения, поэтому следует применить длинную арифметику.

Пример 3. Энты

Энты были созданы в Первоначальную эпоху вместе с другими обитателями Средиземья. Эльфийские легенды гласят, что когда Варда зажгла звёзды и пробудились Эльфы, вместе с ними пробудились и Энты в Великих Лесах Арды.

Когда Энты пришли в Арду, они ещё не умели говорить — этому искусству их обучали Эльфы, и Энтам это ужасно нравилось. Им доставляло удовольствие изучать разные языки, даже щебетание Людей.

Эльфы выработали хорошую технику обучения энтов своему языку. Первый энт, которого обучили эльфы, выучил всего два слова — «tancave» (да) и «la» (нет). Обученный энт выбрал одного старого и одного молодого энта, не умеющих говорить, и обучил их всем словам, которые знал сам. Затем обучение этих двух энтов продолжили сами эльфы. Каждый обучившийся у эльфов энт снова выбирал из неговорящих сородичей одного старого и одного молодого, обучал их всем словам, которые знал, передавал эльфам и так далее.

Выяснилось, что более молодые энты выучивали у эльфов ещё ровно столько же слов, сколько они узнали от обучавшего их энта. А вот более старые, уже склонные к одеревенению энты, пополняли свой запас всего лишь одним словом. После обучения у эльфов энты до конца света уже не могли выучить ни одного нового слова.

Общее число энтов в Средиземье больше, чем вы думаете. Интересно, а сколько из них знают ровно 150 квенийских слов? Похожую задачу вам предстоит решить.

Входные данные:

Входной файл INPUT.TXT содержит натуральные числа K и P ($K \leq 10^6$; $1 \leq P \leq 10^9$), записанные через пробел.

Выходные данные:

Мы понимаем, что число энтов, знающих в точности K слов, может быть слишком велико, поэтому просим вывести в выходной файл OUTPUT.TXT лишь количество энтов, знающих ровно K слов, по модулю P(Табл. 9).

Талица 9

Примеры входных и выходных данных «Энты»

№	INPUT.TXT	OUTPUT.TXT
1	4 10	2
2	8 10	5
3	360 1000	179

Задача имеет динамическое решение. Среди тех, кто будет знать ровно i слов будут как молодые, так и старые Энты, каждый из них обучался у какого-то Энта и впоследствии у Эльфов. Старые Энты, знающие ровно i слов, могли обучиться этому только у Энтов, которые знали ровно i-1 слово (еще одно слово они получают от Эльфов). Молодые Энты получат i слов от тех, кто знал ровно i/2 слов (вторую половину они получают от Эльфов). Заметим, что после обучения молодые Энты всегда знают четное количество слов, поэтому их следует учитывать только для четных i. Поэтому если a[i] – количество Энтов, знающих ровно i слов, то a[i]=a[i-1], если i нечетно и a[i]=a[i-1]+a[i/2] иначе. Поскольку первый Энт знал 2 слова, то a[2]=1, логично также, что a[0]=a[1]=0, т.к. обученных Энтов с нулевым или одним словом нет. Увеличивая значение i от 3х до k в результате в a[k] мы получим ответ. Стоит так же заметить, что каждый раз, определяя очередное значение,

следует вычислять остаток отделения на p, а не один раз это делать в конце программы. Это позволит избежать переполнение целого типа.

Изучив условия задач, сравнив их с классическими задачами решаемыми методом динамического программирования описанными в разделе 1.1 нашей квалификационной работы, мы прийдем к выводу, что хотя бы 1 из заданий олимпиады возможно решить методом динамического программирования. Это значит, что учащемуся профильного класса необходимо давать основы метода динамического программирования.

1.5 Анализ существующих обучающих материалов по динамическому программированию

Для обучения решению задач методом динамического программирования существует множество отрывочных лекций. Изучив все доступные материалы как на физическом носителе(книги), так и в сети интернет, нам удалось собрать 3 достойных курса, которые затрагивают тему динамического программирования не полноценно, но в достаточном объеме для изучения основ. Рассмотрим три лучших образовательных ресурса.

1. Он–лайн школа «Фоксфорд»
2. Сайт Константина Полякова
3. Учебник для школьников Станислава Михайловича Окулова

Последовательно разберем их плюсы и минусы.

Он–лайн школа «Фоксфорд»

Онлайн-школа основанная на площадке видео YouTube(Рисунок 9).

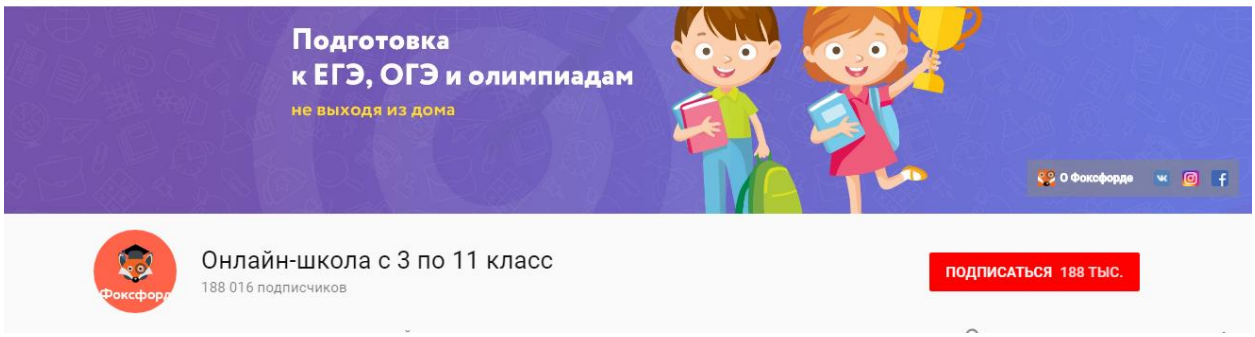

Рисунок 9. Фоксфорд

Курс включает в себя все задания ОГЭ, ЕГЭ, а также дополнительные задания для расширения границ знаний в определенных, специализированных областях. Главным плюсом курса, является его легкодоступность, бесплатность, объемность информации. В куре рассмотрены как основы динамического программирования, так и часть классических задач.

Минусами является недостаточность материала, аудиальный и визуальный вид представления информации, отсутствие дополнительного материала, обратной связи.

Сайт Константина Полякова

На данном сайте материал по методу динамического программирования представлен в виде двух различных решений(Рис. 10):

1. Учебное пособие углубленного уровня
2. Материалы задач представленных в ЕГЭ

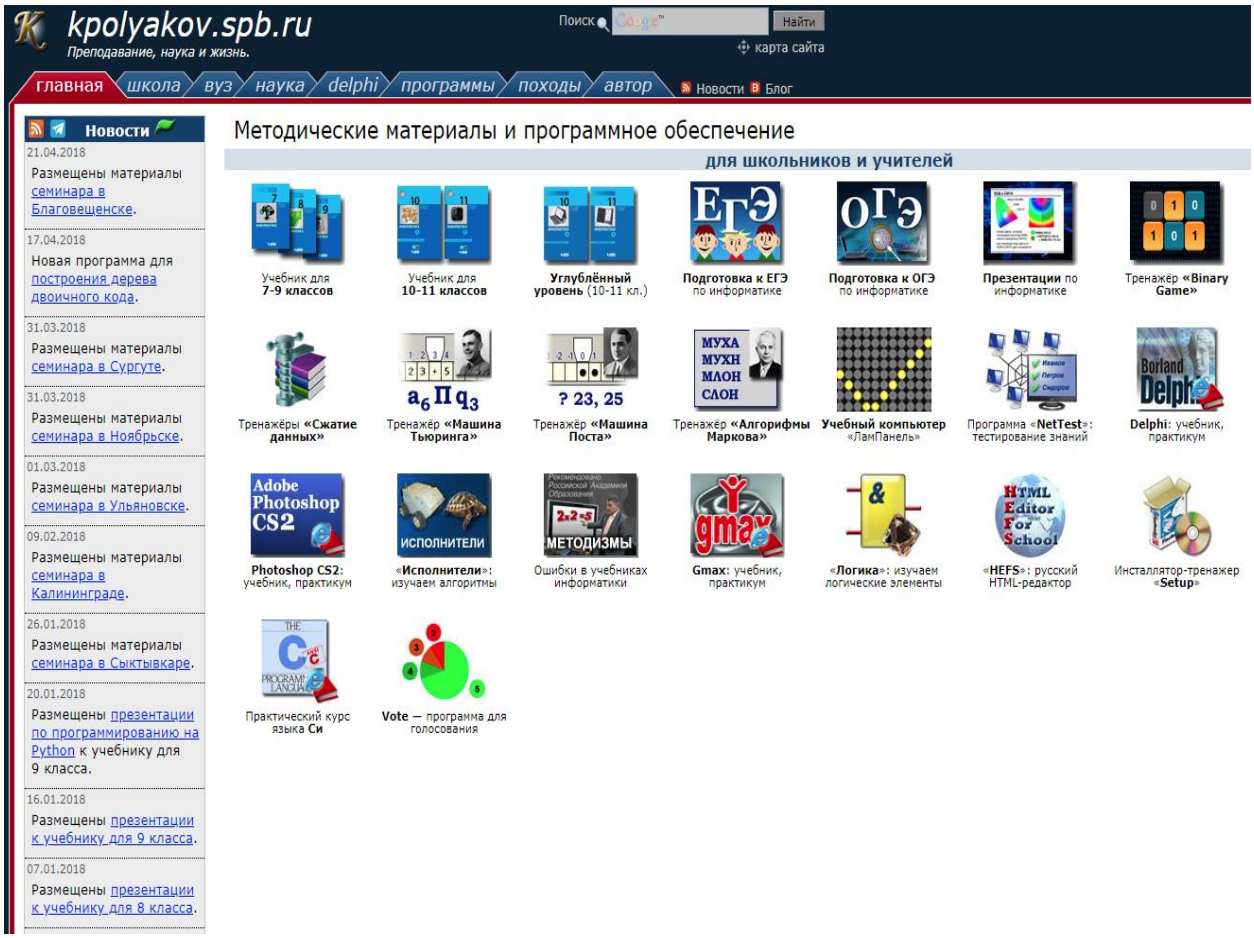

Рисунок 10. Сайт Константина Полякова

Учебное пособие «Информатика.11 кл. Углубленный уровень. Часть 2.» для 11 классов содержит широкий перечень тем, которые разработаны и упрощены для школы и помогают будущим абитуриентам с легкостью определиться с выбором специализации обучения. К сожалению тема динамического программирования рассматривается лишь частично в учебнике, занимая 1 параграф книги(Рис 11.). Отсутствие полного разбора классических задач не дадут возможность учащемуся полноценно решать различные ситуационные задачи методом динамического программирования. В учебники разобраны две задачи 15,22 ЕГЭ решенные методом динамического программирования.

Глава 6. Алгоритмизация и программирование 49
§ 38. Целочисленные алгоритмы 49
§ 39. Структуры (записи) 57
§ 40. Динамические массивы 66
§ 41. Списки 73
§ 42. Стек, очередь, дек 82
§ 43. Деревья 95
§ 44. Графы 107
§ 45. Динамическое программирование 119

Рисунок 11. Параграф «Динамическое программирование»

Задачи ЕГЭ рассмотрены более подробно. В данных файлах разбирается несколько вариантов решения задач в том числе и методом динамического программирования. Но к сожалению подробность описания не даст готовящемуся учащемуся полноту знаний по теме динамическое программирование(Рис. 12).

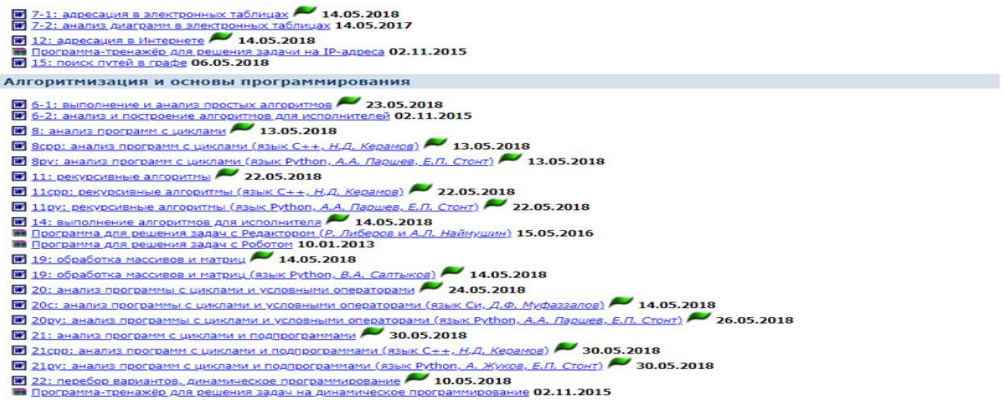

Рисунок 12. Задачи ЕГЭ

Учебник для школьников Станислава Михайловича Окулова

Главным недостатком изучения метода динамического программирования по данному учебному пособию является то, что в открытом доступе присутствует лишь часть учебника. 17 страниц включающих в себя введение в динамическое программирование, решение задачи последовательность Фибоначчи на этом заканчивается открытость материала.

К сожалению уровень изложения поднят до уровня высшего образования и сложен в понимании даже для школьников профильного уровня. Отсутствие методических материалов, готовых решений уроков и части классических задач, которые часто встречаются на олимпиадах

Выводы по I главе

В данной главе были рассмотрены основные понятия, история возникновения, виды задач с конкретными примерами, проведен анализ курсов, уроков, материалов в котором представлена информация о методе динамического программирования.

Проанализированы часто используемые ресурсы, как материальные, так и представленные в сети интернет, которыми чаще всего пользуются учителя. Ни один из источников не упрощен для более простого понимания учащимися. Теме динамическое программирование уделяется недостаточно времени.

Также были изучены материалы - задачи олимпиад 2015-2018. Из материалов видно, что интерес к теме «динамическое программирование» постоянен. С 2015 на олимпиадах всегда присутствует 1 задача, как и в 2018 году которую можно решить методом динамического программирования.

Как результат рассмотренных в первой главе материалов служат крепким фундаментом для теоретических основ, формирования и наполнения элективного курса «Динамическое программирование».

Глава 2. Разработка элективного курса «Динамическое программирование»

2.1 Анализ нормативных документов

Неотъемлемой частью разработки любой образовательной программы в современной России является требования федерального образовательного стандарта(ФГОС)[17]. Требования обязательны к выполнению при создании образовательных программ, методических рекомендаций, элективных и других курсов. Информатика, как обязательный школьный предмет вводится на уровне основного общего образования, который для старших классов подразделяется на базовый и профильный уровень. Далее будет рассматриваться этот уровень с профильной направленностью.

Содержание элективного курса, уроков должно разрабатываться в соответствии с требованием ФГОС И основной образовательной программой основного общего образования (ООП ООО)[15]. В ФГОС для информатики выделенная специальная предметная область «Математика и информатика», куда входит информатика. Все предметные результаты отражены в отдельном параграфе 11.3. Планируемые результаты освоения курса определяют ведущие цели, основу, суть на которой строится вся образовательная программа курса. Они помогают лучше понять, чего достигнет и чему научится ученик после завершения образовательной программы.

Описание основных требований и содержания учебного предмета «Информатика приводится в отдельном параграфе». Реализация данной программы способствует развитию универсальных учебных действий, которые отражены в ФГОС. Важной частью ООП ООО является описание и особенностью организации учебной деятельности с целью развития ИКТ-компетенции учащихся в образовательном процессе. Также ООП ООО направлена на достижение планируемых результатов, развитие личности

учащегося основываясь на психолого-педагогических возрастных особенностях развития учащихся[16].

Кроме этого в стандарте выделяется раздел отражающий требования к результатам освоения образовательной программы, который соединяется с требования ООП ООО.

Соединив требования ФГОС и ООП, можно выделить три группы результатов освоения образовательной программы:

1. **Личностные.**

Направлены на формирование ценностно-смысловых установок, личностной и гражданской позиции, способности к самостоятельному развитию.

2. **Предметные.**

Являются более узконаправленными среди всех результатов и касаются конкретного учебного предмета. В них включается изучение предметных понятий, практическое применение знаний, умение оперировать знаниями в различных ситуациях.

3. **Метапредметные.**

Включают в себя освоение межпредметных понятий и достижение универсальных учебных действий. В метапредметных результатах отмечается развитие способности к самостоятельному планированию действий, а также контроля за их достижением.

Поняв, что именно отражают метапредметные, личностные и предметные результаты в элективном курсе, подберем подходящие из представленного в ФГОС списка по методу динамического программирования, либо схожие с ним:

1. Личностные

- Формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в

мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов, а также на основе формирования уважительного отношения к труду, развития опыта участия в социально значимом труде;

- Формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;

2. Метапредметные

- Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- Умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- Умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- Умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
- Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- Умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи,

строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

3. Предметные

- Понимание роли информационных процессов в современном мире;
- овладение простейшими способами представления и анализа статистических данных; формирование представлений о статистических закономерностях в реальном мире и о различных способах их изучения, о простейших вероятностных моделях; развитие умений извлекать информацию, представленную в таблицах, на диаграммах, графиках, описывать и анализировать массивы числовых данных с помощью подходящих статистических характеристик, использовать понимание вероятностных свойств окружающих явлений при принятии решений;
- Развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера, пользоваться оценкой и прикидкой при практических расчётах;
- Формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- Формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойствах;
- Развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами — линейной, условной и циклической;

- Формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей — таблицы, схемы, графики, диаграммы, с использованием соответствующих программных средств обработки данных;

Изучив образовательный стандарт, можно прийти к выводу, что он включает в себя лишь основные умения, которые необходимы для развития школьников и становления в своей профессиональной деятельности. Упоминание динамического программирования полностью отсутствует. Задача элективного курса развить дополнительные умения не включенные в общую образовательную программу.

2.2 Элективный курс «Динамическое программирование»

Метод динамического программирования используется при решения современных олимпиадных задач, в заданиях ЕГЭ. Однако, отсутствуют методические разработки обучения динамическому программированию школьников. На данный момент в общеобразовательной школе (в том числе и в профильных классах) выделяется недостаточно времени на изучение динамического программирования. Современная литература и курсы рассчитаны на готового специалиста, разбирающегося в предмете и основах динамического программирования. Таким образом, разработка методики обучения динамическому программированию в школьном курсе информатики актуальна.

По данной теме был разработан элективный курс «Динамическое программирование»

Элективный курс – это повышение квалификации для школьников, расширение их области знаний все дальше и дальше в область узкоспециализированной деятельности.

В основе данного курса лежат все достоинства остальных рассмотренных в первой главе онлайн курсов, учебных пособий. Пусть в

курсе собраны все достоинства по сравнению с другими, не стоит отрицать, что есть и недостатки.

Собрав все знания и данные необходимые для курса «Динамическое программирование», стоит перейти к непосредственному их соединению в материалах курса, уроков. Эти знания лежат в основе курса формируя систему базовых знаний, понятий и представлений о динамическом программировании.

Новизна курса состоит в уже закрепленных знаниях программирования, создания алгоритмов, реализации их для решения практических задач. Новая тема динамического программирования, поможет решать группу задач качественно находя решения без внутренних ошибок исполнения, когда при обработки более 1 миллиона строк документа необходимо выделить некоторое решение по заданному условию, что простым перебором занимает займет несколько часов, где методом динамического программирования задача будет решена за час – полтора

Элективный курс предназначен как для внеурочной деятельности, так и включения в общую образовательную программу внутри школы. Программа курса была составлена с учетом возрастных особенностей учащихся и рассчитана с учетом использования вычислительной техники в компьютерном классе за персональными компьютерами[13]. Курс предложен для учащихся 10-11 профильных классов продолжительностью 13 часов. Занятия проводятся в группах 10-20 человек, рассчитанные на занятия длительностью 1 час 1 раз в неделю.

Цель курса – сформировать у учеников представление о методе динамического программирования. Научить решать задачи методом динамического программирования и описывать их на языке Паскаль.

Данная цель достигается решениями следующих *задач*:

1. Развить у учащихся основные навыки и умения решения задач методом динамического программирования;

2. Формировать у учащихся представление об информационной деятельности человека и информационной этике как основах современного информационного общества;
3. Развивать познавательные, интеллектуальные и творческие способности обучающихся, выработать навыки применения метода динамического программирования в практической деятельности, при выполнении индивидуальных и коллективных проектов, дальнейшем освоении профессий, востребованных на рынке труда.

Планируемые результаты освоения курса

Сформулированная цель реализуется через достижение образовательных результатов. Эти результаты структурированы по ключевым задачам дополнительного общего образования, отражающим индивидуальные, общественные и государственные потребности, и включают в себя личностные, предметные, метапредметные результаты.

Личностные результаты:

- Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- Формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики;
- Развитие осознанного и ответственного отношения к собственным поступкам при работе с информацией;
- Формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской, творческой и других видов деятельности.

Предметные результаты:

- Умение использовать термины «динамическое программирование», «подзадача», «оптимизация», «алгоритм», «программа»,

«Рекуррентная формула», «Перекрывающиеся подзадачи», «Оптимальная подструктура»;

- Умение решать задачи 15,22 из ЕГЭ методом динамического программирования;
- Умение составлять рекуррентные формулы для задач решаемых методом динамического программирования;
- Записывать решение задач методом динамического программирования на выбранном алгоритмическом языке (языке программирования);
- Умение формально выполнять алгоритмы задач решаемых методом динамического программирования.

Метапредметные результаты:

- Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- Умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
- Умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
- Смысловое чтение;
- Умение осознанно использовать речевые средства в соответствии с задачей коммуникации; владение устной и письменной речью.

- Увеличение объема информации повышает роль информатики в школьном образование. Создание опытных специалистов максимально просвященных в своей профессиональной деятельности, квалифицированных кадров, которые так необходимы для экономики государства, научно-технического прогресса. Именно это и отражает актуальность данного курса.

Кроме основных результатов элективного курса «Динамическое программирование» необходимо разработать темы уроков и обозначить их время реализации – тематическое планирование(Табл. 10).

Таблица 10

Тематическое планирование курса.

Тема	Количество часов		
	Всего	Теория	Практика
Введение в динамическое программирование	1	1	0
Основные понятия, виды задач, решаемых методом динамического программирования	1	1	0
История создания метода Беллмана	1	1	0
Количество путей в графе. Задача ЕГЭ № 15	1	0,5	0,5
Исполнитель калькулятор. Задача ЕГЭ № 2	1	0,5	0,5
Задачи ЕГЭ 15,22	1	0	1
Задача о мячике	1	0,2	0,8
Задача о лягушках	1	0,2	0,8
Одномерные задачи динамического программирования	1	0,2	0,8

Двумерное задачи динамического программирования	2	1	1
Сложные олимпиадные задачи	1	0	1
Решение задач методом динамического программирования	1	0	1
Итого:	13	5,6	7,6

В курсе представлены материалы как для ученика, так и для учителя. Материалы для учителя отражают методические рекомендации к урокам. В задачи учителя будет входить лишь дополнительная доработка материала под стиль преподавания, доработка визуальной составляющей для повышения интереса и мотивации учащихся, расширение преподаваемых материалов в зависимости от обученности учащихся и новизны материалов.

Все уроки разработаны на основе ФГОС и ООП ООО отраженных в пояснительной записке к элективному курсу

В данном разделе содержатся методические рекомендации по проведению 13 уроков специально разработанных для учителя. Также в разделе содержится пояснительная записка к элективному курсу. В ней описывается: цель, задачи, планируемые результаты освоения курса (личностные, предметные, метапредметные). Материалы не однотипны и отличаются от урока к уроку. Всего можно выделить пять типов урока: теоретический урок, знакомство с методом решения конкретных примеров задач, самостоятельные работы, разбор олимпиадных и ЕГЭ задач.

Ниже представлены методические рекомендации по проведению уроков.

Урок № 1 «Введение в динамическое программирование»

Планируемые образовательные результаты:

Предметные

Умение использовать термины «динамическое программирование», «подзадача», «оптимизация», «алгоритм», «программа», «Рекуррентная формула», «Перекрывающиеся подзадачи», «Оптимальная подструктура»;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение понимания основ метода динамического программирования;

2) Создание связей между методом решения задач перебором, линейно и динамически;

3) Оценка знаний и практических навыков учащихся при решении задач на языке программирования Паскаль.

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Код.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер.

Особенности изложения содержания темы урока

Правильное распределение времени и четкие ответы на вопросы учеников о основах метода динамического программирования. В основном стоит уделить внимание на решение задачи «последовательность Фибоначчи»

Урок № 2 «Основные понятия, виды задач, решаемых методом динамического программирования»

Планируемые образовательные результаты:

Предметные

Умение использовать термины «динамическое программирование», «подзадача», «оптимизация», «алгоритм», «программа», «Рекуррентная формула», «Перекрывающиеся подзадачи», «Оптимальная подструктура»;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение понимания основ метода динамического программирования;

2) Создание связей между методом решения задач перебором, линейно и динамически;

3) Оценка знаний и практических навыков учащихся при решении задач на языке программирования Паскаль.

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Классические задачи; Этапы решения задач.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер.

Особенности изложения содержания темы урока

Следует уделить особое внимание этапам решения задачи. Для лучшего усвоения материала ученики должны самостоятельно составить алгоритм решения основываясь на этапах решения задачи методом динамического программирования.

Урок № 3 «История создания метода Беллмана»

Планируемые образовательные результаты:

Предметные

Умение формально выполнять алгоритмы задач решаемых методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение понимания основ метода динамического программирования;

2) Создание связей между методом решения задач перебором, линейно и динамически;

3) Оценка знаний и практических навыков учащихся при решении задач на языке программирования Паскаль.

Особенности изложения содержания темы урока

Основой данного урока служит последовательность разработки методов решения математических задач. Должны быть рассмотрены 4 метода решения вплетающиеся в историю для одной задачи. Используйте презентацию грамотно и если необходимо будет сменить задачу для разбора, следует прорешать ее с учениками тремя различными алгоритмами одним из которых будет метод динамического программирования.

Урок № 4 «Количество путей в графе. Задача ЕГЭ № 15»

Планируемые образовательные результаты:

Предметные

Умение решать задачи 15 из ЕГЭ методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной

деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Граф; Ветви; Вершины.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер.

Особенности изложения содержания темы урока

Основной упор в данном уроке должен быть сделан на последовательном разборе одной задачи с последующим решением схожих по условиям. Чем больше задач решат ученики методом, тем качественней будет освоен материал урока.

Урок № 5 «Исполнитель калькулятор. Задача ЕГЭ № 22»

Планируемые образовательные результаты:

Предметные

Умение решать задачи 22 из ЕГЭ методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм;

Подзадачи; Исполнитель калькулятор; Действия.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Основной упор в данном уроке должен быть сделан на последовательном разборе одной задачи с последующим решением схожих по условиям. Чем больше задач решат ученики методом, тем качественней будет освоен материал урока.

Урок № 6 «Задачи ЕГЭ 15,22»

Планируемые образовательные результаты:

Предметные

Умение решать задачи 15,22 из ЕГЭ методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке: нет.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Данный урок – это самостоятельная работа по двум прошедшим задачам 15,22 из ЕГЭ. Стоит уделить особое внимание разбору ошибок учеников.

Урок № 7 «Задача о мячике»

Планируемые образовательные результаты:

Предметные

Умение решать задачу «задача о мячике» методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Задача о мячике: Входные и Выходные данные; Код.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

В начале урока осуществляется:

1) Повторение основных понятий: Динамическое программирование, рекуррентная формула, алгоритм, подзадачи, код. — использование 2 слайда электронного приложения к уроку — презентации «Урок № 7»

2) Постановка условия задачи «Задача о мячике» — использование 3 слайда электронного приложения к уроку — презентации «Урок № 7»

Далее нужно подробно разобрать пример решения задачи раскрытый в уроке. Алгоритм и последовательность вопросов, которые помогают решить и понять решение задачи.

Урок № 8 «Задача о лягушке»

Планируемые образовательные результаты:

Предметные

Умение решать задачу «задача о лягушке» методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной

деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Задача о лягушке; Входные и Выходные данные; Код.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

В начале урока осуществляется:

1) Повторение основных понятий: Динамическое программирование, рекуррентная формула, алгоритм, подзадачи, код. — использование 2 слайда электронного приложения к уроку — презентации «Урок № 7»

2) Постановка условия задачи «Задача о мячике» — использование 3 слайда электронного приложения к уроку — презентации «Урок № 7»

Далее нужно подробно разобрать пример решения задачи раскрытый в уроке. Алгоритм и последовательность вопросов, которые помогают решить и понять решение задачи.

Урок № 9 «Одномерные задачи динамического программирования»

Планируемые образовательные результаты:

Предметные

Умение решать задачу «Одномерные задачи динамического программирования» методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Одномерное динамическое программирование; Ячейка; Входные и Выходные данные; Код.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Основное внимание стоит уделить роботе и созданию массива в котором даны и будут хранится данные.

Урок № 10-11 «Двумерное динамическое программирование»

Планируемые образовательные результаты:

Предметные

Умение решать задачу «Двумерное динамическое программирование» методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке:

Динамическое программирование; Рекуррентная формула; Алгоритм; Подзадачи; Двумерное динамическое программирование; Робот; Входные и Выходные данные; Код.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Основное внимание стоит уделить роботу и полю по которому он передвигается. Решить задачу с роботом методом перебора и показать насколько невыгодно, так решать. Стоит рассмотреть дополнительно задачи лабиринт, если останется время.

Урок № 12 «Олимпиадные задачи»

Планируемые образовательные результаты:

Предметные

Закрепить умение решать олимпиадные задачи методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке: нет.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Для лучшего усвоения и понимания олимпиадных задач учащимися, следует разобрать и проанализировать одну из предложенных в уроке задач.

Используя Olympcode.h1n.ru можно проверить решения задач с подобранными под задачи тестами.

Урок № 13 «Решение задач методом динамического программирования»

Планируемые образовательные результаты:

Предметные

Проанализировать ошибки учащихся при решение задач методом динамического программирования;

Метапредметные

Умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной

деятельности, развивать мотивы и интересы своей познавательной деятельности;

Владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

Личностные

Формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

Решаемые учебные задачи:

1) расширение представлений о сферах использования динамического программирования;

2) Использование знаний об алгоритме решения задач методом динамического программирования

Основные понятия, рассматриваемые на уроке: нет.

Используемые на уроке средства ИКТ:

Персональный компьютер (ПК) учителя;

Мультимедийный проектор;

Экран;

Персональный компьютер;

Особенности изложения содержания темы урока

Последний урок служит оценкой всего элективного курса, поэтому стоит подобрать задачи, которые будут показывать умение учеников, которые они приобрели за весь курс. Список задач может быть различен, но структуру в примере карточки необходимо сохранить. Используя Olympcode.h1n.ru или любой другой ресурс с базой данных задач, можно составить свои варианты для учеников.

2.3 Программно-методическая поддержка элективного курса

В качестве программно-методической поддержки элективного курса «Динамическое программирование» был разработан сайт. Сайт разработан на основе WordPress. Элективный курс располагается на сайте «http://a96308sg.beget.tech».

На главной странице сайта расположено меню, которое состоит из 4-х вкладок: «Элективный курс», «Материал для учителя», «Материал для ученика», «Контакты».

Вкладка «Элективный курс» - главная страница сайта на которой располагается вся информация о элективном курсе: тематическое планирование, о проекте, имя создателя сайта, цель программы, планируемые образовательные результаты(Рис. 13).

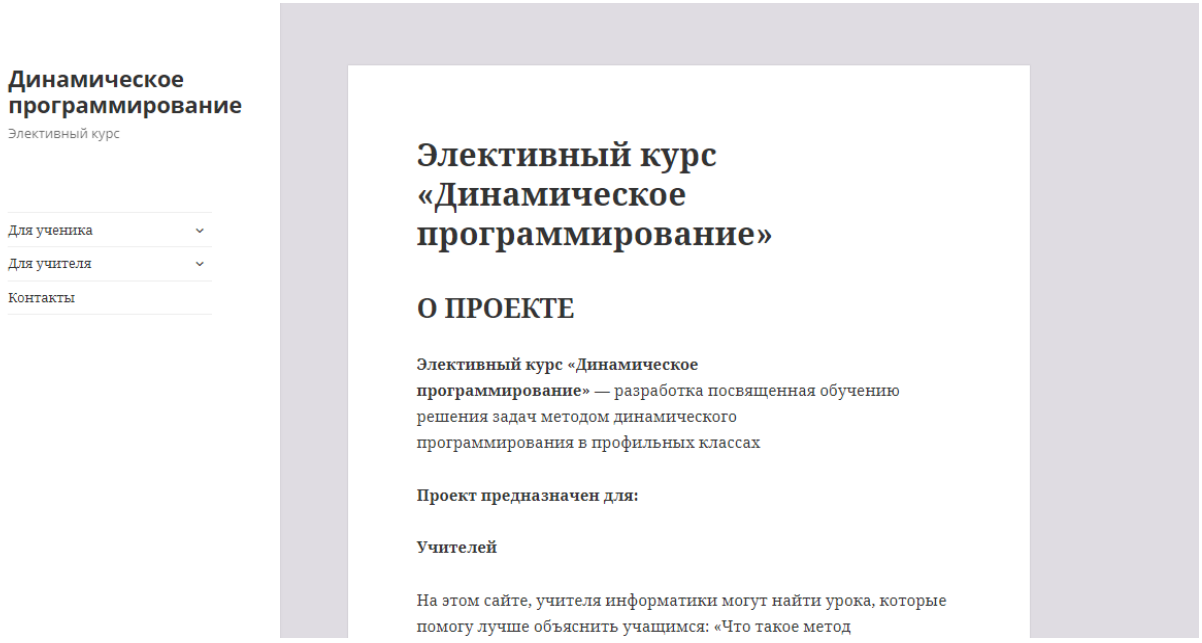

Рисунок 13. Страница элективного курса

Элементы главного меню «Материал для учителя», «Материал для ученика» содержат выпадающее меню с 13 пунктами уроков разработанных для элективного курса «Динамическое программирование».

Каждый урок в разделе «Материал для ученика» отражает содержание урока. Материалы не однотипны и отличаются от урока к уроку. Всего можно выделить пять типов урока: теоретический урок, знакомство с методом решения конкретных примеров задач, самостоятельные работы, разбор олимпиадных и ЕГЭ задач(Рис. 14).

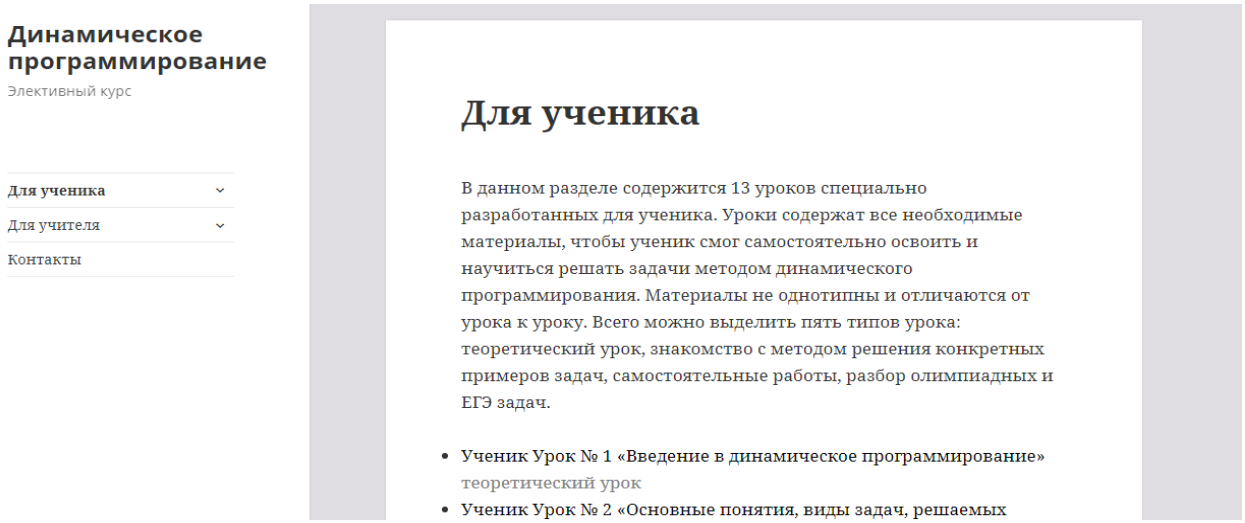

Рисунок 14. Материал для ученика

В уроках типа: знакомство с методом решения конкретных примеров задач решаемых методом динамического программирования также содержатся задачи повышенного уровня сложности(Рис. 15).

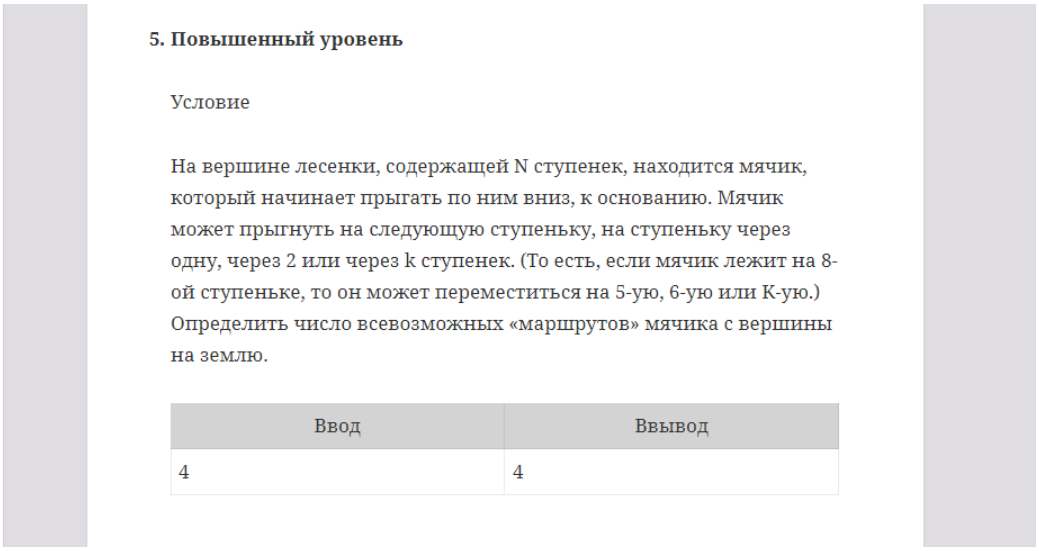

Рисунок 15. Задачи повышенного уровня сложности

В разделе «Материал для учителя» представлена презентация и методические рекомендации для проведения урока(Рис. 16), самостоятельная работа для ученика с помощью которой можно контролировать понимание и проверять пройденный на уроке материал, в некоторых уроках содержится необходимый дополнительный материал: ЦОР для повышения мотивации и интереса к уроку у учеников, повышения качества восприятия и запоминания информации, наглядности.

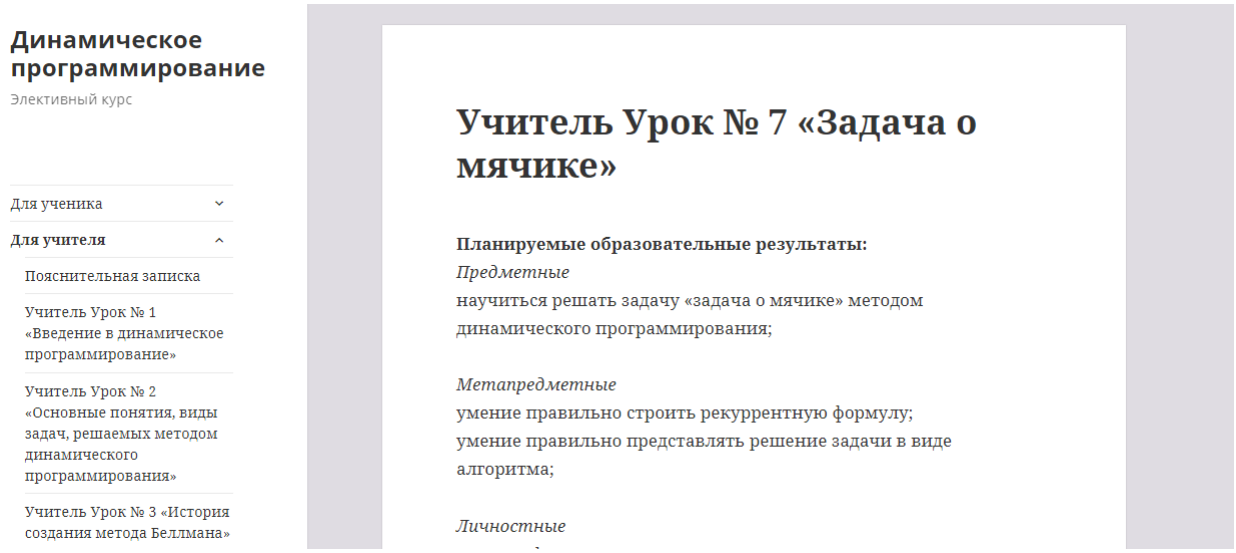

Рисунок 16. Методические рекомендации

Также в подменю «Материал для учителя» содержится пояснительная записка к элективному курсу. В ней описывается: цель, задачи, планируемые результаты освоения курса (личностные, предметные, метапредметные)(Рис. 17).

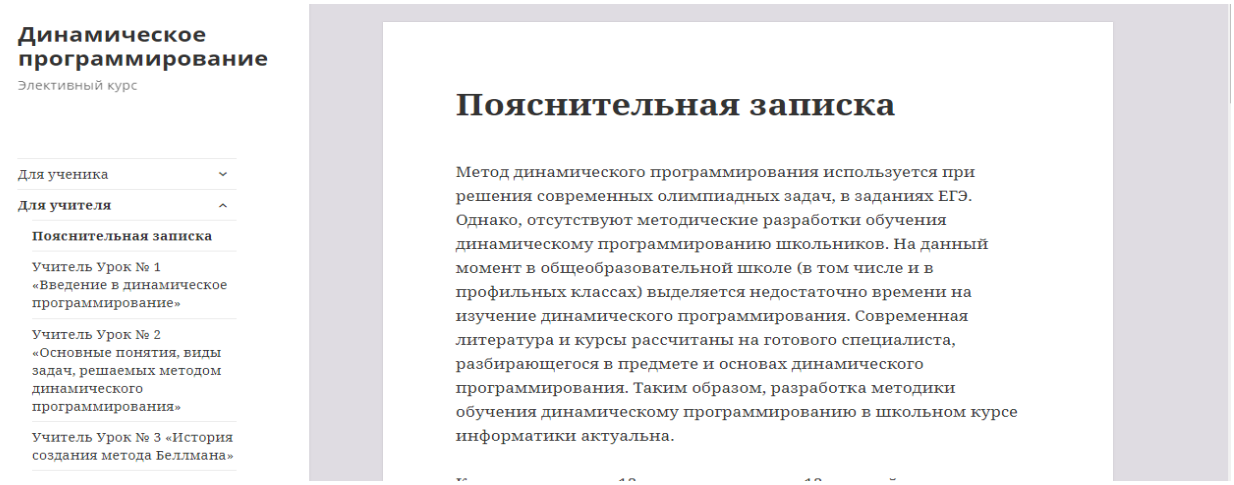

Рисунок 17. Пояснительная записка

2.4 Апробация результатов исследования в школе

Педагогическая апробация проводилась во время научно-исследовательской педагогической практики в МАОУ СОШ № 67. г. Челябинска. Уроки были проведены в 10 классе. В течении двух занятий были рассмотрены темы:

1. Введение в динамическое программирование. Основные понятия, виды задач – 1 час.
2. Задача о мячике – 1 час.

Апробация элективного курса прошла успешно. Способствовало этому правильная мотивация, цели и задачи изучения темы, грамотно подобранный простой и понятный материал первого теоретического урока. За счет правильно объясненной теории и знанию построения простых алгоритмов, ученики с легкостью поняли суть метода и около 2/3 класса смогли самостоятельно и легко реализовать код программы второй практической задачи. Наибольший интерес у учеников вызвал второй урок из-за использования простой и понятной инфографики процесса решения задачи.

Выводы по II главе

На основе теоретических положений, сравнений курсов, изложенных в первой главе, во второй главе представлено описание элективного курса «Динамическое программирование» для учеников профильных классов, включающего программно-методическую поддержку в виде электронного пособия выполненного в виде сайта для учеников и учителей.

В методической поддержке к урокам для учителя представлены презентации с полным раскрытием темы, которые можно сразу использовать на уроке.

Апробация курса проводилась в рамках педагогической практики в МАОУ СОШ № 67. г. Челябинска. На занятиях ученики 11 классов с удовольствием выполняли задания, активно отвечали на вопросы. Таким образом, во второй главе исследования был разработан и апробирована часть элективного курса «Динамическое программирование» и программно-методическая поддержка к нему.

ЗАКЛЮЧЕНИЕ

Подводя итоги моей квалификационной работы, следует отметить, что проведенное исследование направлено на изучение метода динамического программирования, разработку элективного курса с программно-методической поддержкой. Актуальность исследования обусловлена тем, что сегодня информационные технологи широко распространены в мире. Знания элективного курса «Динамическое программирование» поможет ученикам не только при сдаче ЕГЭ, а так же личном развитии.

В процессе исследования были выполнены поставленные задачи и получены следующие результаты:

1. Изучены теоретические положения по проблеме исследования, в школьном курсе данная тема не рассматривается.
2. Разработан элективный курс «Динамическое программирование» на 13 часов для профильных классов.
3. Разработана программно-методическая поддержка элективного курса «Динамическое программирование» в виде сайта.

В подтверждение гипотезы можно сказать, что данный курс позволяет повысить уровень знаний учащихся профильных классов по теме «метод динамического программирования». Таким образом, поставленные задачи можно считать выполненными, также можно сделать вывод о верности поставленной гипотезы.

Библиографический список

1. Андреева Е.В. Еще раз о задачах на полный перебор вариантов[Текст]/ Е.В. Андреева - “Информатика”. - 2000. - №45.
2. Ахо А.А. Структуры данных и алгоритмы [Текст]/ А.А. Ахо, Д.Э. Хопкрофт, Д.Д. Ульман. - М.: Вильямс. - 2007. – 400 с
3. Беллман Р. Прикладные задачи динамического программирования [Текст]/ Р. Беллман, С. Дрейфус. - М.: Наука - 1965. - 458 с.
4. Гордеев Э.Н. Задачи выбора и их решение. [Текст]/ Э.Н. Гордеев. - В кн.: Компьютер и задачи выбора. - М.: Наука - 1989.
5. Демоверсии, Кодификатор [Электронный ресурс]/ Федеральная служба по надзору в сфере образования и науки, Федеральное Государственное Бюджетное Научное Учреждение, «Федеральный институт педагогических измерений». – Электрон. Текстовые данные. – Режим доступа: http://http://fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory - 16.06.2018
6. Демоверсии, Спецификатор [Электронный ресурс]/ Федеральная служба по надзору в сфере образования и науки, Федеральное Государственное Бюджетное Научное Учреждение, «Федеральный институт педагогических измерений». – Электрон. Текстовые данные. – Режим доступа: http://http://fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory - 16.06.2018
7. Окулов С. М. Динамическое программирование [Текст]/ С. М. Окулов, О. А. Пестов. – М. : БИНОМ. Лаборатория знаний, 2012. – 296 с. – 978-5-9963-2572-6
8. Поляков К. Информатика. [Текст]: 11 класс. Углубленный уровень: учебник в 2 ч. /К.Ю. Поляков, Е.А. Еремин. – М.: БИНОМ. Лаборатория знаний, 2012. – 312 с.
9. Кормен, Т., Алгоритмы: построение и анализ [Текст]: учебник /

Т.Х. Кормен, Ч. И. Лейзерсон, Р.Л. Ривест, К. Штайн. - М.: МЦНМО - 2000. – 960 с.

10. Кормен, Т., Алгоритмы: построение и анализ [Текст]/ Т.Х. Кормен, Ч. И. Лейзерсон, Р.Л. Ривест, К. Штайн. - 2-е изд. — М.: Издательский дом "Вильямс", 2005. - 1296 с.

11. Кормен, Т., Динамическое программирование [Текст]/ Т.Х. Кормен, Ч. И. Лейзерсон, Р.Л. Ривест, К. Штайн. - 2-е изд. — М.: Вильямс - 2005. — 1296 с

12. Кузнецов А.В Математическое программирование [Текст]: Учеб. пособие для эконом. спец. вузов / А.В Кузнецов, Н.И. Холод. – Мн.: Выш. шк. - 1984. – 221 с.

13. Обухова, Л. Ф. Возрастная психология [Текст]/ учебное пособие / Л. Ф. Обухова. — М.: Юрайт, 2011. — 460 с.

14. Окулов С.М. Задачи по программированию [Электронный ресурс]: С. М. Окулов [и др.]; под ред. С. М. Окулова. — 2-е изд., испр. (эл.). — М. : БИНОМ. Лаборатория знаний, 2014. — 823 с. - 978-5-9963-2372-2 – Режим доступа: http://files.lbz.ru/pdf/cC2372-2-ch.pdf

15. Примерная основная образовательная программа основного общего образования от 8.04.2015 г. [Электронный ресурс] - https://минобрнауки.рф/документы/922

16. Статья на тему: Психологические особенности обучающихся 9-11 классов [Электронный ресурс] - http://www.studfiles.ru/preview/1721703 - Дата обращения: 16.04.2017 г.

17. Федеральный государственный образовательный стандарт основного общего образования от 17.12.2010 г. №1897 [Электронный ресурс] – http://минобрнауки.рф/документы/543

18. Щербина О. А. Методологические аспекты динамического программирования [Текст]/ О.А. Щербина. - Динамические системы. Межведомственный научны сборник. – Выпуск 22 - 2007.