

DOI: 10.32517/2221-1993-2024-23-5-85-94

В. С. Попов

Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет), г. Москва, Россия;

Тверской государственный университет, г. Тверь, Россия

Е. А. Алефиренко, Л. Ю. Черницына

Московский городской педагогический университет, г. Москва, Россия

РЕКУРСИВНАЯ ФУНКЦИЯ ДЛЯ РЕШЕНИЯ ЗАДАНИЯ НА ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ: ОТ БЕЗЗНАНИЕВОЙ ФОРМЫ КОМПЕТЕНЦИИ К ЗНАНИЮ И ПОНИМАНИЮ

Аннотация

Рекурсивная функция для решения задания 23 ЕГЭ по информатике на динамическое программирование используется экзаменуемыми без глубокого понимания принципов ее работы: при сформированном навыке ее использования отсутствуют необходимые знания, что является беззнанием формы компетенции. Формирование понимания принципов работы программного кода и переход от беззнания формы к полноценным компетенциям и компетентностям — информационным, цифровым, математическим, а также в области программирования — позволит ученикам получить более глубокие знания теории рекурсивных функций, сформировать навыки, деятельностные и ценностные отношения в области решения задач динамического программирования на более высоком уровне, что способно привести к улучшению образовательных результатов, пониманию методов динамического программирования, применяемых в том числе на ЕГЭ и олимпиадах по информатике. Для формирования знаниевой компоненты компетенции предлагается построение, изучение и использование деревьев вызовов рассматриваемой рекурсивной функции, созданных в среде программирования Observable, а также изучение мемоизации как ключевого отличия метода динамического программирования от рекурсии.

Ключевые слова: динамическое программирование, динамическое программирование в школе, рекурсия, рекурсивная функция, дерево рекурсивных вызовов, информационные модели на графах, компетенция, компетентность, беззнаниевая компетенция.

Контактная информация

Попов Владислав Сергеевич,

старший преподаватель кафедры «Информационные системы и телекоммуникации», факультет «Информатика и системы управления», Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет), г. Москва, Россия; *адрес:* 105005, Россия, г. Москва, 2-я Бауманская ул., д. 5, стр. 1;

аспирант Института педагогического образования и социальных технологий, Тверской государственный университет, г. Тверь, Россия; *адрес:* 170100, Россия, г. Тверь, ул. Желябова, д. 33;

e-mail: popov_vlad@mail.ru

Алефиренко Евгения Александровна, старший методист (проектная группа «Информатика в предпрофессиональном образовании»), Институт развития профильного обучения, Московский городской педагогический университет, г. Москва, Россия; *адрес:* 129226, Россия, г. Москва, 2-й Сельскохозяйственный проезд, д. 4, корп. 1; *e-mail:* alefirenkoea@mgpu.ru

Черницына Лариса Юрьевна, методист (проектная группа «Информатика в предпрофессиональном образовании»), Институт развития профильного обучения, Московский городской педагогический университет, г. Москва, Россия; *адрес:* 129226, Россия, г. Москва, 2-й Сельскохозяйственный проезд, д. 4, корп. 1; *e-mail:* chernicynalyu@mgpu.ru

V. S. Popov

Bauman Moscow State Technical University, Moscow, Russia;

Tver State University, Tver, Russia

E. A. Alefirenko, L. Yu. Chernitsyna

Moscow City University, Moscow, Russia

RECURSIVE FUNCTION FOR SOLVING A DYNAMIC PROGRAMMING TASK: FROM KNOWLEDGELESS FORM OF COMPETENCE TOWARDS KNOWLEDGE AND UNDERSTANDING

Abstract

The recursive function for solving task 23 on dynamic programming of the Informatics Unified State Exam is used by examinees without a deep understanding of the principles of its operation: with the formed skill of its use, the necessary knowledge is missing, which is an knowledgeless form of competence. Understanding the principles of the program code and the transition from the knowledgeless

form to full-fledged competencies (information, digital, mathematical, programming) will allow students to gain deeper knowledge of the theory of recursive functions, to form skills and attitudes in the field of solving dynamic programming problems at a higher level, which can lead to improved educational results, understanding of dynamic programming methods used, including Unified State Exam tasks and informatics Olympiads. To form the knowledge component of the competence, it is proposed to build, study and use call trees of the recursive function, created in the Observable programming environment, as well as study memoization as a key difference between the dynamic programming method and recursion.

Keywords: dynamic programming, dynamic programming at school, recursion, recursive function, recursion tree, graph information models, competence, competency, knowledgeable competence.

1. Введение

1.1. Задания на вычисление количества программ

Задание на вычисление количества программ преобразования одного числа из другого при заданных командах исполнителя появилось в части С демонстрационного варианта ЕГЭ по информатике 2012 года [6]:

«У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 1,
2. умножь на 3.

Первая из них увеличивает число на экране на 1, вторая — утраивает его. Программа для Утроителя — это последовательность команд. Сколько есть программ, которые число 1 преобразуют в число 29? Ответ обоснуйте».

Большое количество (по нашим субъективным оценкам, более половины) выпускников, принимавших участие в ЕГЭ по информатике в компьютерной форме (КЕГЭ), выполнило бы данное задание с использованием программного кода, использующего рекурсивную функцию (листинг 1).

Некоторые из выпускников, справившихся с подсчетом количества программ, преобразующих число 1

в число 29, привели бы следующее обоснование, показывающее количество программ для каждого возможного числа в траектории вычислений (листинг 2).

Данный программный код выведет количество способов получения очередного числа i из 1, называемых в терминологии задания программами, для каждого i от 1 до 29 включительно (см. табл.).

Впрочем, оператор `return f(x + 1, y) + f(x * 3, y)` сам по себе является обоснованием полученного ответа через использование рекуррентной формулы $f(x, y) = f(x + 1, y) + f(x * 3, y)$ и соответствующей базы рекурсии: в соответствии с критериями демонстрационного варианта ЕГЭ по информатике 2012 года за верную рекуррентную формулу в ответе можно было получить один балл из трех [6, с. 29–31], а за формулу и вычисления, содержащие арифметические ошибки, — два балла.

В задании 22 демонстрационного варианта ЕГЭ по информатике 2016 года впервые вводится термин «траектория вычислений программы», а само задание несколько усложняется [7]:

«Исполнитель Май15 преобразует число на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Умножить на 2

```
def f(x, y):
    if x > y:
        return 0
    if x == y:
        return 1
    if x < y:
        return f(x + 1, y) + f(x * 3, y)
print(f(1, 29)) # ответ: 23
```

Листинг 1. Решение задания С3 демонстрационного варианта ЕГЭ 2012 года в Python

```
def f(x, y):
    if x > y:
        return 0
    if x == y:
        return 1
    if x < y:
        return f(x + 1, y) + f(x * 3, y)
for i in range(1, 30):
    print(i, f(1, i))
```

Листинг 2. Решение задания С3 демонстрационного варианта ЕГЭ 2012 года с выводом количества программ для всех чисел траектории вычислений в Python

Таблица

**Числа и количество программ для их получения из 1
в задании С3 демонстрационного варианта ЕГЭ 2012 года**

Число	Кол-во программ	Число	Кол-во программ	Число	Кол-во программ
1	1	11	5	21	15
2	1	12	7	22	15
3	2	13	7	23	15
4	2	14	7	24	18
5	2	15	9	25	18
6	3	16	9	26	18
7	3	17	9	27	23
8	3	18	12	28	23
9	5	19	12	29	23
10	5	20	12		

Первая команда увеличивает число на экране на 1, вторая умножает его на 2. Программа для исполнителя Май15 — это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 29 и при этом траектория вычислений содержит число 14 и не содержит числа 25? Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17».

В наши дни большинство экзаменуемых выполнили бы данное задание с помощью дополненного программного кода (листинг 3).

Поскольку число 25 не входит в траекторию вычислений, при $x = 25$ рекурсивная функция f возвращает значение 0. Поскольку число 14 входит в траекторию вычислений, количество программ для получения числа 29 из числа 2 равняется произведению количества программ $f(2, 14)$ на количество программ $f(14, 29)$.

Рассматриваемое задание на динамическое программирование, появившись в демонстрационном варианте ЕГЭ 2012 года под номером С3, занимало различные позиции в перечне заданий ЕГЭ по информатике: в 2013 и 2014 годах — В13, в 2015–2020 годах — 22, в 2021–2024 годах — 23. Фактически современное задание 23 ЕГЭ по информатике присутствует в ЕГЭ с 2016

по 2023 год и соответствует приведенному заданию демонстрационной версии ЕГЭ по информатике 2016 года.

1.2. Актуальность, проблема, объект и предмет исследования

Объектом исследования в данной статье является рекурсивная функция для решения задания на динамическое программирование 23 ЕГЭ по информатике. Предмет исследования: программные и графические средства перехода от беззнаний формы компетенции к пониманию принципов работы рекурсивной функции для решения задания на динамическое программирование 23 ЕГЭ по информатике в контексте компетентностного подхода. Объект и предмет исследования рассмотрены с позиций компетентностного подхода в образовании, использующегося в Федеральных рабочих программах по информатике для X–XI классов базового и углубленного уровней [34, 35].

Хотя многие обучающиеся обладают умением, навыком и даже ценностным отношением в контексте использования представленного программного способа решения заданий на динамическое программирование посредством рекурсии, далеко не все они смогут объяснить организацию и принципы работы рассмотренного программного кода. На ЕГЭ по информатике задание на динамическое программирование в 2019–2023 годах верно было решено примерно половиной

```
def f(x, y):
    if x > y or x == 25:
        return 0
    if x == y:
        return 1
    if x < y:
        return f(x + 1, y) + f(x * 2, y)
print(f(2, 14) * f(14, 29)) # ответ: 13
```

Листинг 3. Решение задания 22 демонстрационного варианта ЕГЭ 2016 года в Python

выпускников [26], что свидетельствует о среднем уровне выполнения данного задания экзаменуемыми как по отношению к рассмотренной задаче, так и в сравнении с прочими видами заданий экзамена. Впрочем, в ЕГЭ по информатике 2015 года данное задание верно выполнили лишь 25,6 % выпускников [18, с. 31]. В методических рекомендациях для учителей, подготовленных на основе анализа типичных ошибок участников ЕГЭ по информатике 2015 года, указано, что данное задание повышенного уровня сложности 22 выполняется с низким, но в целом приемлемым результатом [18, с. 8]. Как можно судить по [18, 26], дальнейший опыт решения данного задания позволил увеличить средний процент выполнения до 50 %. Однако последующая практика перехода от ЕГЭ к КЕГЭ подтверждает довольно парадоксальный вывод статьи [25] о том, что появившиеся при введении КЕГЭ в 2021 году разнообразные способы решений и проверки заданий не приводят к существенному улучшению образовательного результата, выраженного в числовом виде в результате педагогического измерения на экзамене. Школьники, получив навык решения различных заданий посредством программного кода, но не понимая ни особенностей работы данного кода, ни возможностей аналитического решения задания, фактически проявляют беззнаниевую форму компетенции [2; 3, с. 165; 4, с. 95; 17; 36, с. 12], в которой применение, владение и практический навык подавляют знаниевую компоненту [там же]. Формирование понимания [1, 22] принципов работы программного кода и переход от беззнаниевой к полноценной компетенции и компетентности [15, с. 23; 29, с. 58, 62; 30] позволяют ученикам получить более глубокие знания теории рекурсивных функций, сформировать навыки и ценностные отношения в области решения задач динамического программирования, что способно привести к улучшению результатов решения задач, использующих методы динамического программирования, в том числе на ЕГЭ и олимпиадах по информатике. На вопросы: «Как это работает?» и «Почему это работает?» мы постараемся ответить в данной статье, что позволит перейти от беззнаниевой компетенции к знанию и пониманию — по крайней мере, в отношении обозначенных объекта и предмета исследования. В «Педагогическом энциклопедическом словаре» понимание рассматривается как «мыслительный процесс, направленный на выявление существенных свойств предметов и явлений действительности, познаваемых в чувственном и теоретическом опыте человека» [22, с. 206–207]. Е. Г. Белякова рассматривает проблему понимания с различных позиций и подходов [1], расширяя трактовку термина на основе современного герменевтического, культурологического и педагогического знания. Процессы понимания и познания, расширение знаний субъекта о мире, свойствах предметов и явлений в нем оказывают наибольшее влияние именно на знаниевую компоненту компетенций.

Многие авторы публикаций по теме решения задач на динамическое программирование в школьном курсе информатики не рассматривают внутрипредметные и межпредметные связи, необходимые для формирования полноценного понимания методов решения данного задания, фокусируясь на конкретных способах и методиках решения. Напротив, пример реализации

интегративного подхода с использованием укрупненных дидактических единиц описывают Н. И. Заводчикова, И. А. Быкова [13]. Д. Е. Томме, М. В. Поспелов рассматривают взаимодействие учителей математики и информатики для эффективного обучения предметному и междисциплинарному содержанию [31]. Тем не менее рассмотрение различных способов, методов и методик решения задания на динамическое программирование, по нашему мнению, также приводит к улучшению понимания задания обучающимися, формированию знаний и компетенций. А. С. Муртузалиева, М. М. Халимбеков [20], А. Н. Ерофеев [12] приводят аналитическое табличное решение заданий, опирающееся на рекуррентные соотношения, что связано с «аналоговой» формой проведения ЕГЭ по информатике вплоть до 2020 года. З. М. Муцурова в работе [21] приводит популярное до появления компьютерного ЕГЭ решение с использованием рекуррентных формул и обратного хода рекурсии. Е. Е. Рябинина в своей статье [28] также использует два предыдущих метода при решении заданий на динамическое программирование. И. И. Каширская рассматривает использование деревьев и обратных деревьев для решения задания [14]. С. В. Козлов, А. А. Быков описывают решение задания в электронных таблицах [16], а Т. В. Шамаева демонстрирует применение электронных таблиц на практике [38]. Н. А. Долгушин, М. Н. Оленькова в статье [10] дают программный циклический способ решения еще до появления компьютерного ЕГЭ по информатике. А. С. Трекина, М. В. Богданова рассматривают рекурсивное решение задания на языках программирования Python и Pascal [32, с. 350–351]. Е. П. Шафоростова описывает аналитический табличный и программный циклический способы решения задания [39]. Наконец, А. А. Лысыкова и Е. А. Редько в работе [19] рассматривают несколько способов решения задания, включая аналитический табличный, аналитический графический, рекурсивный и циклический, не рассматривая, однако, задания с ограничениями в траектории вычислений. А. А. Чапурных в статье [37] демонстрирует использование таблиц, рекуррентных формул и рекурсии на языках программирования Python и Pascal. Н. Е. Гаенко указывает на недостаточное количество часов и отсутствие методических разработок для обучения динамическому программированию школьников, необходимость расширения понимания основ метода динамического программирования [5, с. 585]. А. В. Тришкина подчеркивает необходимость формирования информационных компетенций и устойчивого интереса к профессиональной деятельности в области информационных технологий обучающихся технологического профиля, потребность в предпрофессиональном обучении [33, с. 167]. В методических рекомендациях для учителей, подготовленных на основе анализа типичных ошибок участников ЕГЭ 2015 года, рассматриваются низкий процент выполнения данного задания, а также умение использовать динамическое программирование различными группами учащихся [18]. Дальнейшие методические рекомендации для учителей, подготовленные на основе анализа типичных ошибок участников ЕГЭ 2016–2023 годов, не содержат подробных описаний решения рассматриваемого задания на динамическое про-

граммирование, указывая только процентный результат выполнения данного задания участниками экзамена*. Наблюдается противоречие между, с одной стороны, необходимостью формирования информационных, цифровых, математических компетенций, глубокого понимания обучающимися методов динамического программирования и, с другой стороны, сложившимися образовательными практиками их изучения и применения, а также недостатком учебной и методической литературы по данной теме.

Еще одним подтверждением актуальности представленной работы является ситуация, сложившаяся в связи с переходом на новый формат аттестации обучающихся на уровне среднего общего образования: ЕГЭ по информатике как форма государственной итоговой аттестации во многом стал определять предметное содержание. Ранее в содержании учебного предмета «Информатика» динамическое программирование отсутствовало, но затем введение и рост популярности ЕГЭ по информатике [27] привели к тому, что задания экзамена, ранее не изучаемые в школьном курсе информатики, стали его предметным содержанием. В демонстрационном варианте ЕГЭ по информатике 2024 года методами динамического программирования могут быть решены задания 18 (исполнитель Робот в лабиринте), 19–21 (теория игр), 23 (рассматриваемое задание динамического программирования), 27 (задание ЕГЭ на программирование самого высокого уровня сложности) [9, 24]. В демонстрационном варианте ЕГЭ по информатике 2023 года метод динамического программирования был применен к заданию 22 (многозадачность) [8]. До внедрения КЕГЭ по информатике методы динамического программирования также использовались при решении заданий на вычисление количества решений систем логических уравнений. Таким образом, более пятой части заданий современного ЕГЭ по информатике могут быть решены методами динамического программирования, которые, как следствие, следует изучать как на углубленном, так и на базовом уровнях изучения учебного предмета «Информатика».

2. Методы

В нашем исследовании были использованы следующие научные методы:

- методы программирования: приведены и рассмотрены программные коды для решения задания на динамическое программирование 23 ЕГЭ по информатике, приведены программы для построения дерева рекурсивных вызовов;
- методы теории графов: для рассмотренного задания построены графы — полное и неполное деревья рекурсивных вызовов, иллюстрирующие принципы работы рассматриваемых рекурсивных функций;
- метод анализа научной и педагогической литературы.

* См. методические рекомендации для учителей, подготовленные на основе анализа типичных ошибок участников ЕГЭ за 2016–2023 годы, В. Р. Лещинера, М. А. Ройтберга, С. С. Крылова.

3. Результаты и обсуждение

3.1. Дерево рекурсивных вызовов

Вычисление ответа рекурсивной функцией, приведенной в листинге 1 для задания С3 может быть визуализировано в виде дерева рекурсивных вызовов (рис. 1), на примере которого следует формировать знаниевый компонент компетенции при решении рассматриваемого задания. В случае получения целевого числа 29 (выполняется условие базы рекурсии $x = y$) функция f возвращает значение 1, все такие случаи выделены на рисунке 1 и приведены отдельно на рисунке 2. В случае получения числа x , большего 29 (выполняется условие базы рекурсии $x > y$), рекурсивная функция f возвращает значение 0. Ответ 23, полученный в листинге 1, равен количеству путей от числа 1 (корневой узел) до числа 29 (подмножество концевых узлов) на графе или количеству листьев дерева со значением 29 (см. рис. 1). Необходимо сформировать понимание того, что именно это количество и вычисляется рекурсивной функцией листинга 1. Для деревьев небольшого размера и высоты возможно построение дерева рекурсивных вызовов вручную.

Программный код для визуализации дерева рекурсивных вызовов на языке JavaScript в среде Observable [41] приведен в листинге 4.

Каждая строка массива *ways* листинга 4 соответствует одному пути и представляет собой разделенные слешами узлы дерева. Значение 1 в корне дерева соответствует начальному числу задания. Каждый лист дерева содержит дополнительную информацию о возвращаемом значении функции (надпись $\rightarrow 0$ соответствует выполнению оператора `return 0` в функции f ; надпись $\rightarrow 1$ соответствует `return 1`). Строки сформированы при помощи видоизмененной рекурсивной функции листинга 1, сохраняющей траекторию вычислений в виде строки и представленной в листинге 5**.

Неполное дерево рекурсивных вызовов, содержащее исключительно пути от корневого узла дерева (1) до терминальных узлов, имеющих значение 29, приведено на рисунке 2***.

Для задания 22 демонстрационного варианта ЕГЭ 2016 года можно построить аналогичные деревья для вызовов $f(2, 14)$ и $f(14, 29)$. Произведение количества единиц, возвращенных функциями $f(2, 14)$ и $f(14, 29)$, является ответом на задание (см. листинг 3).

3.2. О связи динамического программирования и рекурсии

Используя приведенные в листингах 1–3 программы, содержащие для решения задачи динамического программирования рекурсивную функцию, обучающиеся редуцируют задание динамического программирования к получению результата вычисления рекурсивной функции. Однако динамическое программирование ха-

** Программный код листинга 4 и построенное дерево доступны по ссылке:

*** <https://observablehq.com/@vladislav-popov2/recursion-tree-1>
Программный код и построенное дерево доступны по ссылке:
<https://observablehq.com/@vladislav-popov2/recursion-tree-2>

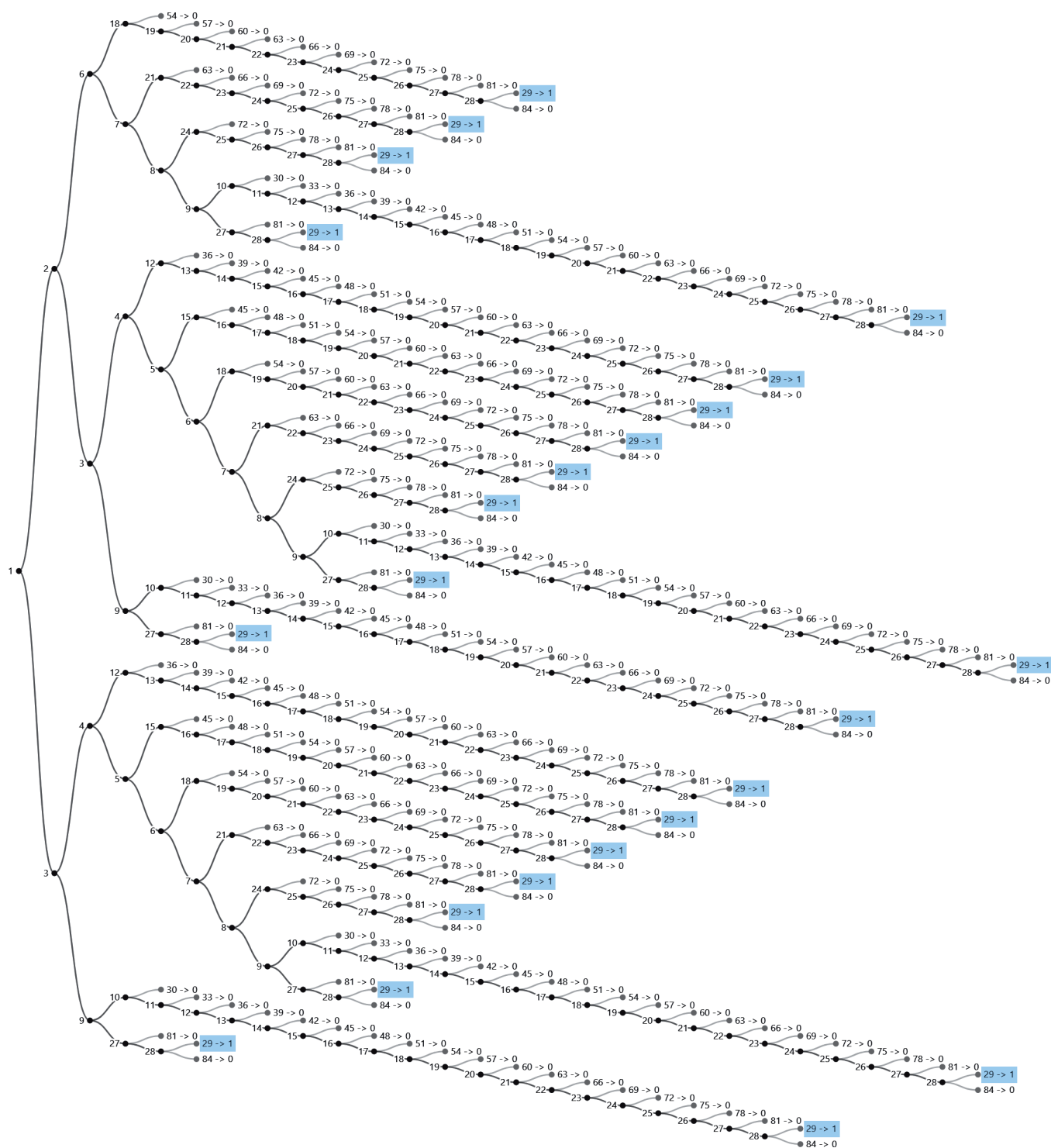


Рис. 1. Полное дерево рекурсивных вызовов для рекурсивной функции, приведенной в листинге 1

характеризуется не только свойством разбиения сложной задачи на более простые и возможностью рекурсивного решения, но также *мемоизацией* — сохранением результатов уже выполненных вычислений, чтобы вычисление значения функции для каждого аргумента выполнялось только один раз [11, 40]. Рассмотренный программный код не содержит мемоизацию, что может быть легко проиллюстрировано деревом рекурсивных вызовов (см. рис. 1) и исправлено применением декоратора *lru_cache* модуля *functools*, позволяющего сохранять уже вычисленные значения функции. Например, при

применении традиционного рекурсивного подхода (см. листинг 1 и рис. 1) значение функции $f(3, 29)$ вычисляется дважды, значение $f(10, 29)$ — уже пять раз, в то время как мемоизация позволит вычислить значение каждой функции только единожды, предоставляя при следующих обращениях к рекурсивной функции с теми же аргументами ранее вычисленное значение. Формирование знаний об отличиях и особенностях динамического программирования и рекурсии также должно являться важным этапом изучения этих необходимых методов вычислений.

```

Plot.plot({
  axis: null,
  width: 1200,
  height: 1200,
  margin: 10,
  marginLeft: 40,
  marginRight: 120,
  marks: [
    Plot.tree(ways, {textStroke: "white"})
  ]
})
ways = [
  "1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29 -> 1",
  "1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/84 -> 0",
  "1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/81 -> 0",
  "1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/78 -> 0",
  ...
]

```

Листинг 4. Программный код для визуализации дерева на языке JavaScript в среде Observable

```

def f(x, y, path):
    if x > y:
        print('\\"' + path + ' -> 0' + '\"' + ',')
        return 0
    if x == y:
        print('\\"' + path + ' -> 1' + '\"' + ',')
        return 1
    if x < y:
        return f(x + 1, y, path + '/' + str(x + 1)) + f(x * 3, y, path + '/' + str(x * 3))
print(f(1, 29, '1')) # после вывода всех путей получен ответ: 23

```

Листинг 5. Видоизмененная программа листинга 1 для вывода всех путей для построения дерева рекурсивных вызовов

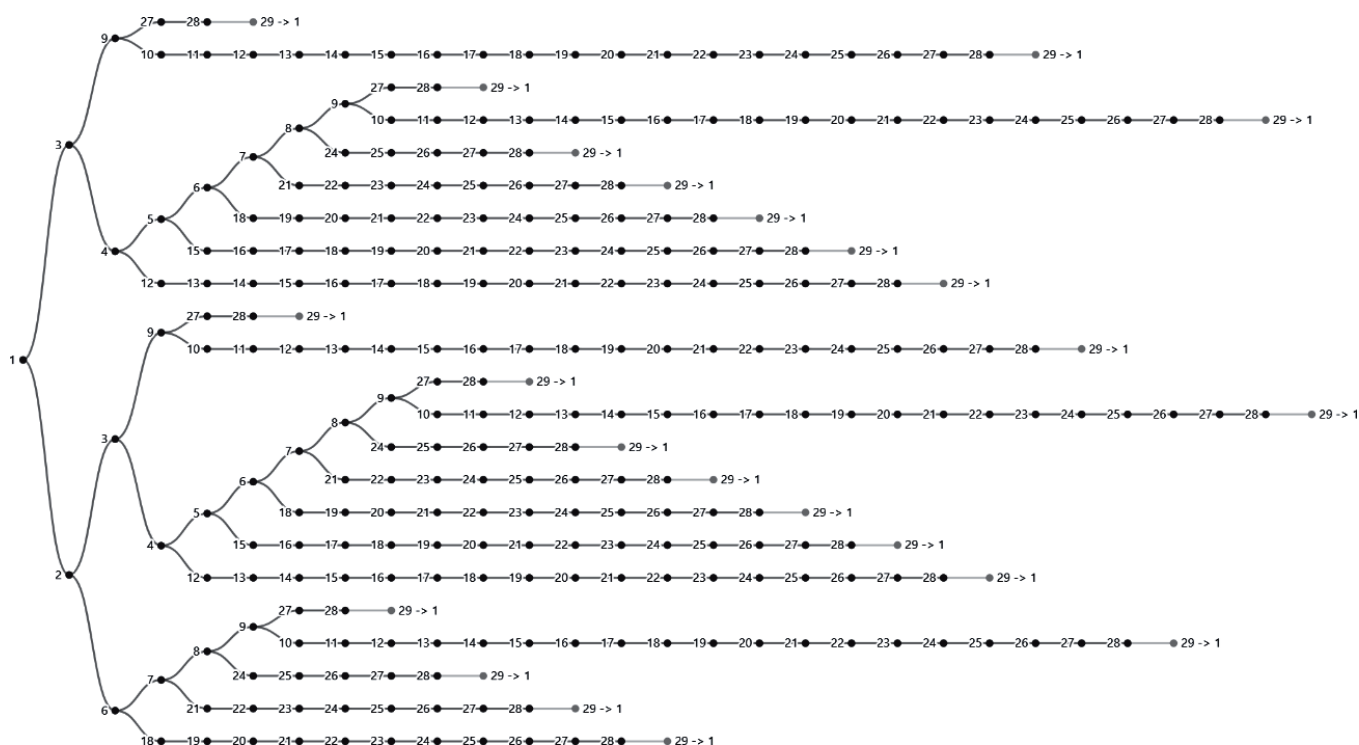


Рис. 2. Неполное дерево рекурсивных вызовов для рекурсивной функции, приведенной в листинге 1 (показаны возможные пути получения из числа 1 числа 29)

3.3. Образовательный результат

Иллюстрация работы рекурсивной функции через построение и изучение дерева рекурсивных вызовов позволяет решить сразу несколько задач:

- сформировать понимание принципа работы широко используемой рекурсивной функции для решения задания динамического программирования в частности и рекурсии в целом, переходя от беззнаний компетенции к формированию всех ключевых компонентов компетенций, включая знаниевый компонент;
- продемонстрировать возможности автоматизированной визуализации на примере построения дерева рекурсивных вызовов в среде Observable;
- сформировать умение автоматизированной визуализации графа с помощью современных средств программирования;
- показать возможности увеличения количества аргументов функции для решения усложненных или дополнительных задач (см. листинг 5);
- на ЕГЭ по информатике 2022 года средний процент выполнения рассматриваемого задания снизился с 47,1 % в 2021 году до 40,6 % [26] ввиду применения целочисленного деления в качестве одной из операций исполнителя, а также большего по значению начального числа, чем конечное число в траектории вычислений (см. задания № 266, 267 в [23]), в связи с чем можно сделать предположение: более глубокое понимание работы рассматриваемой рекурсивной функции может привести к большей способности адаптировать ее код под измененное задание.

4. Заключение

Рассмотренные проблемы при изучении динамического программирования носят комплексный характер, для их решения и достижения высоких образовательных результатов необходима актуализация образовательных ресурсов: в частности, требуется формирование знания и понимания работы рассмотренной в статье рекурсивной функции. Построение и изучение дерева рекурсивных вызовов, его сопоставление с написанной программой помогают сформировать понимание принципов работы широко используемой рекурсивной функции для решения задания динамического программирования, углубить знания о рекурсии. При использовании рекурсивной функции для решения задач динамического программирования также необходимо понимать отличия динамического программирования и рекурсии, главное из которых состоит в сохранении при динамическом программировании результатов уже выполненных вычислений — мемоизации. Реализация предложенного подхода будет особенно полезна в инженерных и ИТ-классах, при реализации программ предпрофессионального образования.

Список источников

1. *Белякова Е. Г.* Понимание как категория смыслоориентированной педагогики // Образование и наука. Известия УрО РАО. 2006. № 6 (42). С. 21–28. EDN: IUZJQV.
2. *Брызгалина Е. В.* Компетентностный подход в системе образования // ПостНаука. https://www.youtube.com/watch?v=lc6EMbZTN_4
3. *Брызгалина Е. В.* Компетентностный подход и будущее социогуманитарного образования // Вестник Волгоградского государственного университета. 2013. № 1 (23). С. 162–168. EDN: QJBGOD.
4. *Брызгалина Е. В.* Проблемы интеграции естественно-научного и философского знания в современном образовании: социальные аспекты // Философия и общество. 2017. № 2 (83). С. 94–96. EDN: ZCPEKN.
5. *Гаенко Н. Е.* Динамическое программирование в школьном курсе информатики // Информационные и инновационные технологии в науке и образовании. Материалы V Всероссийской научно-практической конференции с международным участием. К 65-летию Таганрогского института имени А. П. Чехова (Таганрог, 28–29 октября 2020 года). Ростов н/Д: Издательско-полиграфический комплекс РГЭУ (РИНХ), 2021. С. 584–586. EDN: PHRQBZ.
6. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2012 года по информатике и ИКТ. М.: Федеральный институт педагогических измерений, Федеральная служба по надзору в сфере образования и науки Российской Федерации, 2011. 36 с.
7. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2016 года по информатике и ИКТ. М.: Федеральный институт педагогических измерений, 2015. 54 с.
8. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2023 года по информатике. М.: Федеральный институт педагогических измерений, Федеральная служба по надзору в сфере образования и науки, 2022. 21 с.
9. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2024 года по информатике. М.: Федеральный институт педагогических измерений, Федеральная служба по надзору в сфере образования и науки, 2023. 24 с.
10. *Долгушин Н. А., Оленькова М. Н.* Решение задач ЕГЭ по информатике в среде программирования PascalABC.NET // Международный студенческий научный вестник. 2015. № 3–2. С. 240–241. EDN: TWPKTT.
11. *Долинский М. С.* Ускорение рекурсивных решений при помощи мемоизации // Информатика в школе. 2022. № 6. С. 55–67. EDN: PSWCPK. DOI: 10.32517/2221-1993-2022-21-6-55-67.
12. *Ерофеев А. Н.* Решение задач динамического программирования при подготовке к ЕГЭ по информатике // Актуальные проблемы модернизации математического и естественно-научного образования. Сборник научных трудов по материалам Всероссийской научно-методической конференции (Балашов, 15 мая 2020 года). Балашов: Саратовский источник, 2020. С. 107–110. EDN: RUYALN.
13. *Заводчикова Н. И., Быкова И. А.* Укрупнение дидактических единиц при изучении метода динамического программирования в средней школе // Актуальные проблемы теории и практики обучения физико-математическим и техническим дисциплинам в современном образовательном пространстве. IV Всероссийская (с международным участием) научно-практическая конференция, посвященная 75-летию факультета физики, математики, информатики Курского государственного университета (Курск, 16–17 декабря 2020 года). Курск: Курский государственный университет, 2020. С. 289–294. EDN: GSRACM.
14. *Каширская И. И.* ЕГЭ: задачи 6–2 и 15 как способ подготовки к решению задачи 22 и решение задачи 22 обратным способом // Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции (Воронеж, 14–15 февраля 2019 года). Воронеж: Изд-во «Научно-исследовательские

публикации» (ООО «Вэлборн»), 2019. С. 1840–1845. EDN: ZFDRBB.

15. *Киселева М. М.* Синергетический подход в преподавании информатики как элемент развития ключевых компетенций учащихся // Информатика в школе. 2019. № 4. С. 23–31. EDN: KMDKUV. DOI: 10.32517/2221-1993-2019-18-4-23-31.

16. *Козлов С. В., Быков А. А.* Решение задач динамического программирования ЕГЭ по информатике средствами электронных таблиц // Системы компьютерной математики и их приложения. 2022. № 23. С. 356–361. EDN: ADRMTU.

17. *Коришнуова Н. Л.* Компетентностный и деятельностный подходы к модернизации высшей школы // Лингвокультурное образование в системе вузовской подготовки специалиста. 2017. Т. 1. № 2 (10). С. 184–192. EDN: YSCVIL.

18. *Лещинер В. Р., Ройтберг М. А.* Методические рекомендации для учителей, подготовленные на основе анализа типичных ошибок участников ЕГЭ 2015 года по информатике и ИКТ. М.: Федеральный институт педагогических измерений, 2015. 31 с.

19. *Лысикова А. А., Редько Е. А.* Принцип динамического программирования в решении задания 23 ЕГЭ по информатике // Материалы I региональной студенческой конференции к 65-летию ТОГУ. Материалы конференции (Хабаровск, 27–28 апреля 2023 года). Хабаровск: Тихоокеанский государственный университет, 2023. С. 236–242. EDN: OQYZZV.

20. *Муртузалиева А. С., Халимбеков М. М.* Использование динамического программирования для решения некоторых заданий ЕГЭ по информатике и ИКТ // Наука и образование: состояние, проблемы, перспективы развития. Материалы научной сессии профессорско-преподавательского состава Дагестанского государственного педагогического университета, посвященной 100-летию создания первого в Дагестане педагогического института (Махачкала, 8–9 ноября 2017 года). Махачкала: Дагестанский государственный педагогический университет, 2018. С. 236–239. EDN: RSNJZR.

21. *Муцурова З. М.* Динамическое программирование в задачах ЕГЭ // Известия Чеченского государственного педагогического университета. Серия 2: Естественные и технические науки. 2019. Т. 17. № 1 (20). С. 44–47. EDN: VJYMNW.

22. Педагогический энциклопедический словарь / гл. ред. Б. М. Бим-Бад, редкол.: М. М. Безруких, В. А. Болотов, Л. С. Глебова и др. М.: Большая рос. энцикл., 2002. 527 с.

23. *Поляков К. Ю.* 23: перебор вариантов, динамическое программирование // Сайт Константина Полякова. ЕГЭ по информатике: подготовка к ЕГЭ-2024 по информатике, разбор задач ЕГЭ-2024 по информатике, материалы для подготовки к ЕГЭ. <https://kpolyakov.spb.ru/download/ege23.doc>

24. *Поляков К. Ю.* Динамическое программирование в задачах обработки последовательностей ЕГЭ по информатике // Информатика в школе. 2020. № 5. С. 55–63. EDN: SXZKLG. DOI: 10.32517/2221-1993-2020-19-5-55-63.

25. *Попов В. С.* Анализ изменения среднего процента выполнения заданий ЕГЭ по информатике в связи с введением компьютерного формата экзамена // Новые информационные технологии в образовании. Сборник научных трудов 13-й международной научно-практической конференции (Москва, 31 января — 1 февраля 2023 года). Ч. 2. М.: IC-Паблишинг, 2023. С. 153–156. EDN: DBTYBS.

26. *Попов В. С., Абросимова-Романова Л. А.* Анализ рисков выпускников и выделение необходимых компетенций в связи с изменением среднего процента выполнения заданий ЕГЭ по информатике за 2019–2023 годы и введением компьютерного формата экзамена // Новые информационные технологии в образовании. Сборник научных трудов XXIV Международной научно-практической конференции (Москва, 30–31 января 2024 года). М.: IC-Паблишинг, 2024. С. 326–330. EDN: RJIRBK.

27. *Попов В. С., Видьманов Д. А.* Популярность ЕГЭ по информатике // Преподавание информационных технологий в Российской Федерации. Сборник научных трудов Двадцать

первой открытой Всероссийской конференции (Нижний Новгород, 18–19 мая 2023 года). Н. Новгород: Изд-во Нижегородского государственного университета им. Н. И. Лобачевского, 2023. С. 489–491. EDN: TKPFHZ.

28. *Рябинина Е. Е.* Решение задач ЕГЭ с использованием алгоритмов динамического программирования // Информационно-коммуникационные технологии в педагогическом образовании. 2019. № 6 (63). С. 71–78. EDN: GKGD SA.

29. *Садыкова А. Р., Белоусова А. С.* Методические основы формирования предпрофессиональных ИТ-компетенций старшеклассников в детских технопарках «Кванториум» // Информатика и образование. 2023. Т. 38. № 5. С. 57–64. EDN: LHZVGW. DOI: 10.32517/0234-0453-2023-38-5-57-64.

30. *Самылкина Н. Н.* Структура и содержание цифровых компетенций, формируемых в предпрофессиональном обучении // Информатика в школе. 2020. № 4. С. 11–19. EDN: ZXUPNW. DOI: 10.32517/2221-1993-2020-19-4-11-19.

31. *Томме Д. Е., Поспелов М. В.* О взаимодействии учебных предметов математики и информатики на примере изучения числовых последовательностей // Математика: фундаментальные и прикладные исследования и вопросы образования. Материалы международной научно-практической конференции (Рязань, 26–28 апреля 2016 года). Рязань: Рязанский государственный университет имени С. А. Есенина, 2016. С. 484–486. EDN: XFEPMF.

32. *Трекина А. С., Богданова М. В.* Методические аспекты подготовки учащихся к ЕГЭ по информатике с использованием языка программирования Python // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы международной научно-практической интернет-конференции (Москва, 18–24 апреля 2022 года). М.: МПГУ, 2022. С. 345–352. EDN: WEOGQT.

33. *Тришкина А. В.* Использование различных средств обучения динамическому программированию в ИТ-классах // #ScienceJuice2021. Сборник статей и тезисов (Москва, 22–26 ноября 2021 года). Т. II. М.: ПАРАДИГМА, 2021. С. 167–175. EDN: NGSTYN.

34. Федеральная рабочая программа среднего общего образования. Информатика (базовый уровень) (для 10–11 классов образовательных организаций). М.: Институт стратегии развития образования, 2023. 38 с. https://edsoo.ru/wp-content/uploads/2023/08/21_ФРП-Информатика_10-11-классы_база.pdf

35. Федеральная рабочая программа среднего общего образования. Информатика (углубленный уровень) (для 10–11 классов образовательных организаций). М.: Институт стратегии развития образования, 2023. 52 с. https://edsoo.ru/wp-content/uploads/2023/08/22_ФРП_Информатика-10-11-классы_угл.pdf

36. *Хеннер Е. К.* Профессиональные знания и профессиональные компетенции в высшем образовании // Образование и наука. 2018. Т. 20. № 2. С. 9–31. EDN: NSHCCN. DOI: 10.17853/1994-5639-2018-2-9-31.

37. *Чапурных А. А.* Различные способы решения задания № 23 КЕГЭ на перебор вариантов и динамическое программирование // Наука, общество, образование в условиях цифровизации и глобальных изменений. Сборник статей III Международной научно-практической конференции. Пенза: МЦНС «Наука и Просвещение». 2022. С. 44–48.

38. *Шамаева Т. В.* «Динамическое программирование» с помощью MS Excel на примере 23 номера из КЕГЭ по информатике // Архимед: Научно-методический сборник / отв. за выпуск А. Рубин, Т. Струков, П. Чулков, А. Шевкин. Вып. 18. М.: ООО «МАКС Пресс», 2022. С. 124–136. EDN: RTVFKK. https://www.elemschool.mathedu.ru/text/arhimed_2022_v18/p116/

39. *Шафоростова Е. П.* Динамическое программирование в олимпиадных задачах и ЕГЭ по информатике // Преподавание информационных технологий в Российской Федерации. Сборник научных трудов материалов Двадцатой

открытой Всероссийской конференции (Москва, 19–20 мая 2022 года). М.: 1С-Публишинг, 2022. С. 367–370. EDN: FJKEXE.

40. *Popov V. S.* Dynamic programming in tasks of the Russian Unified State Exam in Computer Science // 2024 6th International

Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2024. P. 1–6. DOI: 10.1109/REEPE60449.2024.10479768.

41. Tree mark | Observable Plot // OBSERVABLEHQ.COM. <https://observablehq.com/plot/marks/tree>