

УДК 372.862:37.04
DOI 10.17513/snt.40187

ОБУЧЕНИЕ АНАЛИЗУ И ВЫЧИСЛЕНИЮ КОЛИЧЕСТВА КОМБИНАЦИЙ ЗАПИСЕЙ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON В ШКОЛЕ

¹Козлов С.В., ²Быков А.А.

¹ФГБОУ ВО «Смоленский государственный университет», Смоленск,
e-mail: svkozlov1981@yandex.ru;

²ФГБОУ ВО «Национальный исследовательский университет «МЭИ»,
Смоленский филиал, Смоленск, e-mail: alex1by@mail.ru

Цель исследования состоит в оценке эффективности объяснения обучающимся методов анализа и вычисления количества комбинаций записей с использованием средств языка программирования Python. Генерация и вычисление количества комбинаций символьных выражений имеет важный прикладной характер в связи с надежным хранением информации в базах данных во избежание в них ошибок и защите от несанкционированного доступа и естественных потерь. Особое значение имеет анализ числовых выражений, записанных с использованием ограничений, которые широко распространены на практике при записи разных идентификаторов. Ввиду этого исследование количества комбинаций возможных записей выражений составляет одну из ветвей дидактической линии «Кодирование информации» школьного курса информатики. При этом выбор оптимальных инструментов решения таких задач является важной составляющей процесса обучения. Так, использование для данных целей средств инструментальных систем открывает перспективы более глубокого изучения программирования в профильных классах. Апробация проводится на двух независимых выборках обучаемых в Смоленском физико-математическом лицее и средней школе № 6 г. Смоленска. Основное внимание уделено использованию встроенных библиотек языка программирования Python и их функций для генерации числовых выражений в разных системах счисления. Это позволяет сгенерировать числовые значения заданной длины, состоящие из указанных значений. Такой подход предоставляет обучающимся механизм формирования комбинаций, среди которых остается выбрать записи, удовлетворяющие условиям задачи. Для этого необходимо воспользоваться алгоритмическими конструкциями ветвления и цикла и применить их композицию при решении задачи. В статье демонстрируются примеры такого использования разного уровня сложности, обсуждаются особенности их программной реализации. Исследовательская работа демонстрирует необходимость овладения навыками организации обработки количества комбинаций выражений средствами языков программирования при обучении профильной информатике в школе.

Ключевые слова: информатика, программирование, комбинаторика, запись, слово, число, алгоритм, язык Python, компьютерная программа, образовательный процесс

LEARNING TO ANALYZE AND CALCULATE THE NUMBER OF COMBINATIONS OF ENTRIES IN THE PYTHON PROGRAMMING LANGUAGE AT SCHOOL

¹Kozlov S.V., ²Bykov A.A.

¹Smolensk State University, Smolensk, e-mail: svkozlov1981@yandex.ru;

²National Research University "MPEI", Smolensk branch, Smolensk, e-mail: alex1by@mail.ru

The purpose of the study is to evaluate the effectiveness of explaining to students the methods for analyzing and calculating the number of record combinations using Python programming language tools. The generation and calculation of the number of combinations of symbolic expressions is of an important application nature in connection with the reliable storage of information in databases in order to avoid errors in them and to protect them from unauthorized access and natural loss. Of particular importance is the analysis of numerical expressions written using constraints, which are widespread in practice when writing different identifiers. In view of this, the study of the number of combinations of possible expression records constitutes one of the branches of the didactic line "Encoding information" of the school computer science course. At the same time, the choice of optimal tools for solving such problems is an important component of the learning process. So, the use of tools of instrumental systems for these purposes opens up prospects for a deeper study of programming in specialized classes. Approbation is carried out on two independent samples of students at the Smolensk Physics and Mathematics Lyceum and secondary school No. 6 in Smolensk. The main focus is on using the built-in libraries of the Python programming language and their functions to generate numerical expressions in different number systems. This allows you to generate numeric values of a given length, consisting of the specified values. This approach provides students with a mechanism for forming combinations, among which it remains to select records that meet the conditions of the task. To do this, you need to use algorithmic constructions of branching and cycle and apply their composition when solving the problem. The article demonstrates examples of such use of different levels of complexity, discusses the features of their software implementation. The research work described by the authors demonstrates the need to master the skills of organizing the processing of the number of combinations of expressions by means of programming languages when teaching specialized computer science at school.

Keywords: computer science, programming, combinatorics, recording, word, number, algorithm, Python language, computer program, educational process

Введение

Вычисление количества комбинаторных комбинаций в школьном курсе информатики относится к теме «Кодирование информации с использованием систем счисления» [1]. Обучающиеся начинают изучать ее в средней школе, когда в курсе математики они знакомятся с темой «Комбинаторика», осмысливают ее основные формулы и получают навыки вычислений количества повторений, размещений и сочетаний [2]. При этом основное внимание уделяется решению заданий, которые требуют подсчета числа вариантов слов или чисел в простейших случаях. В таких задачах достаточно определить число комбинаций знаков на каждой позиции и знать, что итоговый результат получается их перемножением. Решение более сложных задач относится в средней школе к углубленному изучению и выносятся на дополнительные занятия по предмету – элективные курсы и кружки. В старшей школе к этой теме возвращаются на профильном уровне. При этом задания становятся более разнообразными. Они требуют разбора значительно большего количества комбинаций, знаний вычисления вариантов способом от противного, выявления общих и частных закономерностей при анализе возможных ситуаций. Следует также отметить, что возрастает и вычислительная сложность предлагаемых обучающимся заданий [3]. Если при решениях заданий на базовом уровне сложности можно было проводить вычисления «на бумаге», так как ответы редко превосходили четыре тысячи, то на повышенном уровне становилось целесообразно использовать приложение «Калькулятор», так как ответы составляли пятизначные и шестизначные числа. В связи с этим при изучении материала по вычислению комбинаторных комбинаций выражений целесообразно применять современные среды программирования, такие как PyCharm, которые позволяют с помощью инструментов языка Python алгоритмически разрешить поставленную задачу [4]. При этом на получение устойчивых навыков решения таких заданий необходимо отводить больше учебного времени, так как результаты ЕГЭ по информатике свидетельствуют, что обучающиеся в недостаточной мере владеют как навыками формулировки и учета всех условий при решении «вручную», так и навыками их интерпретации в систему команд языка программирования.

Цель исследования – оценка эффективности объяснения обучающимся методов анализа и вычисления количества комбинаций записей с использованием средств языка программирования Python.

Материалы и методы исследования

Решение заданий вычисления количества комбинаций записи чисел «вручную» требует уже в относительно простых примерах разбора множества разных случаев [5, 6]. Так, например, если необходимо определить количество пятизначных семеричных чисел, в записи которых используется не менее трех цифр 6, то перебор вариантов требует достаточно большой аккуратности сначала в их выписывании, а затем в их подсчете. Если же числа задать шестизначные, а цифру 6 заменить на любую нечетную цифру, то число вариантов возрастает, а их учет становится еще более затруднительным занятием. В то же время при использовании языка программирования Python анализ и вычислительная сложность решения этих двух заданий практически не отличаются. Однако применение языка Python также требует базовых знаний и умений в программировании алгоритмических конструкций, таких как ветвление и цикл [7, 8].

Методологический подход к решению таких заданий в среде программирования PyCharm предполагает изучение функционального инструмента `product`, который позволяет организовать перебор чисел в указанной системе счисления заданной длины. Для этого используется цикл с параметром `for` для индексной переменной `i`.

```
for i in product('0123456', repeat = 5):
    s = ''.join(i)
```

Внутри цикла формируется строка `s` указанной длины с помощью значения повторений параметра `repeat`. Строка `s` задается с использованием метода `join`, который «склеивает» ее из символов указанного набора. При этом набор возможных цифр задается там же в конструкции цикла `for` с помощью перечисления цифр в виде строкового значения `'0123456'`.

Рассмотрим подробный пример решения одной из таких задач и разберем его. Пусть необходимо определить количество шестизначных семеричных чисел, запись которых содержит ровно две цифры «5» и на последнем месте не стоит цифра «2».

Программный код решения задачи может выглядеть следующим образом:

```
from itertools import*
k=0
for i in product('0123456', repeat = 6):
    s = ''.join(i)
    if s[0] != '0' and s[-1] != '2' and s.count('5') == 2:
        k+=1
print(k)
```

Ответом на данное задание является число 15336.

Для использования в решении задачи функционального инструмента `product` в программе на языке программирования Python необходимо подключить модуль `itertools`, который содержит методы работы с итерируемыми объектами. Такие объекты позволяют перебирать свои элементы в различном порядке. Употребление после ключевого слова `import` знака «*» означает, что в программе можно будет использовать любые функции подключаемой библиотеки `itertools`. Например, такими функциями могут служить `product` и `permutations`. Отличие этих инструментов при формировании итерируемого объекта состоит в том, что функция `product` позволяет повторять элемент итерации, а функция `permutations` генерирует перечисления без повторений указанной длины. Значение параметра, определяющий длину, элемента выборки задают после ключевого слова `repeat`. В данном примере это значение равно 6. Отметим, что этот параметр не является обязательным. Если его опустить, то длина элемента выборки будет равна длине итерируемого объекта. В рассматриваемом примере количеству цифр в строке `'0123456'` – 7. В случае необходимости использования в программе только конкретных функций подключаемого модуля после ключевого слова `import` их указывают через запятую. Например, `from itertools import product, permutations`. Также отметим, что подключение библиотеки языка программирования Python можно выполнять в любой строке программного кода до момента вызова функции, однако это целесообразно делать в первой строке кода программы.

Затем, для подсчета количества семеричных чисел, запись которых удовлетворяет заданным условиям, определяется начальное значение переменной `k`, равное 0. После этого в цикле `for` с использованием функции `product` с параметрами `'0123456'`, `repeat = 6` перечисляются цифры семеричной системы счисления и указывается количество элемента формируемой выборки. Для дальнейшей обработки полученной последовательности в данной задаче выгодно применить не функцию `list`, а сформировать строку с помощью команды `s = ''.join(i)`. После того как строка сформирована, ее можно проверять на соответствие заданным в условии критериям. Для этого все условия необходимо перечислить в операторе `if`, объединив их логическими функциями `and` и `or` в соответствии с вопросом задания. Если условие в операторе `if` выполняется, то счетчик `k` подходящих числовых значений следует увеличить на 1. Это можно сделать с по-

мощью команды `k+=1`. В завершение программы остается вывести результат вычислений на экран. Для этого необходимо воспользоваться командой `print`, в которой требуется указать имя выводимой переменной для отображения ее значения.

Задание можно усложнить следующим образом. Пусть также необходимо учесть, что рядом не могут стоять две одинаковые цифры.

Решение задания можно представить так:

```
from itertools import*
kk=0
for i in product('0123456', repeat = 6):
    s = ''.join(i)
    if s[0] != '0' and s[-1] != '2' and s.count('5')
== 2:
        k=0
        for j in range(1,5):
            if s[j] == s[j-1]:
                k+=1
            if k == 0:
                kk+=1
print(kk)
```

Ответ на задание – 8575.

Модификация решения состоит в том, что кроме указанных в операторе `if` прежних условий также необходимо учесть, что две одинаковые цифры не могут стоять рядом. Для этого внутри тела оператора `if` следует организовать цикл, в котором сравниваются две соседние цифры. Цикл будет включать пять итераций, на каждой из которых будет проверяться условие равенства значений текущей и предыдущей цифр числа. Если они равны друг другу, то счетчик `k`, который задан до внутреннего цикла `for` и определен начальным нулевым значением, будет увеличиваться на единицу. Иными словами, если в числе есть две рядом стоящие цифры, значения которых совпадают, то он будет отличен от нуля. Если он останется равным 0, то это будет означать, что дополнительное условие задачи выполнено. В этом случае счетчик `kk`, назначенный для подсчета количества чисел, которые удовлетворяют условиям задачи, и инициализированный нулевым значением, будет увеличен на 1. Его значение необходимо вывести в команде `print`, оно будет являться ответом на вопрос задачи.

Такие задания разного уровня сложности решали обучаемые в ходе экспериментальной деятельности. Они образовали методологическую основу исследования. Обработка результатов экспериментальной работы состояла в применении математических методов анализа данных.

Гипотеза исследования заключалась в том, что использование алгоритмических конструкций и инструментов подключаемых библиотек языка программирования Python в школе при обучении анализу и вычислению количества комбинаций записей повышает эффективность образовательного процесса.

Экспериментальная деятельность по обучению подходам к решению задач анализа и вычисления количества комбинаций записей числовых выражений проводилась в Смоленском физико-математическом лицее ЯВИР при МИФИ и МЭИ и средней школе № 6 г. Смоленска. В этих образовательных организациях информатика изучается на профильном уровне. В обучении участвовали 32 ученика.

Результаты исследования и их обсуждение

Результаты педагогического исследования, полученные в ходе диагностических работ, обобщены в табл. 1 и 2.

На констатирующем этапе педагогического эксперимента обучающиеся изучали теоретический материал о способах комбинаторного подсчета возможных ситуаций записи числовых выражений, удовлетворяющих заданным условиям. Они также практиковались решать предложенные по теме «Кодирование информации с использованием систем счисления» задачи «вручную» без использования вспомогательных компьютерных средств, кроме приложения «Калькулятор» для оптимизации вычисле-

ний. Учащиеся применяли при анализе условий заданий знания о записи чисел в различных системах счисления [9, 10], о чередовании четных и нечетных цифр, а также вспоминали основные признаки делимости на числа от 2 до 20. В завершение этого этапа эксперимента обучающимся была предложена диагностическая работа. Она включала семь тестовых заданий разного уровня сложности. Эти задания было необходимо, как и во время практических занятий, решать «на бумаге».

На формирующем этапе педагогического эксперимента обучающиеся тренировали навыки решения таких же задач на компьютере с использованием возможностей языка Python среды программирования PyCharm. В первой части данного этапа они с помощью библиотеки *itertools* и ее инструмента *product* решали задачи базового уровня сложности, предполагавшие в основном цикле формирование чисел в заданной системе счисления и их отбора с использованием оператора ветвления *if*. При этом условия в ветвлении носили составной характер. Обучающимся было необходимо грамотно определить порядок их выполнения с использованием логических функций *and* и *or*. Во второй части этого этапа обучающиеся должны были освоить навыки применения вложенного цикла в основную конструкцию цикла *product*. Это обуславливало необходимость попарного сравнения рядов стоящих цифр анализируемого числа. Такие задания относились к повышенному и высокому уровням сложности.

Таблица 1

Результаты констатирующего этапа педагогического эксперимента

Группа	Число обучающихся, достигших уровня усвоения знаний и сформированности умений			Всего
	Высокий	Повышенный	Базовый	
СФМЛ ЯВИР при МИФИ и МЭИ	4	6	8	18
Средняя школа № 6 г. Смоленска	2	6	6	14
Всего	6	12	14	32

Таблица 2

Результаты формирующего этапа педагогического эксперимента

Группа	Число обучающихся, достигших уровня усвоения знаний и сформированности умений			Всего
	Высокий	Повышенный	Базовый	
СФМЛ ЯВИР при МИФИ и МЭИ	8	8	2	18
Средняя школа № 6 г. Смоленска	6	6	2	14
Всего	14	14	4	32

В завершение этапа обучающимся также была предложена итоговая работа, которая представляла собой диагностический тест. Тест состоял из семи заданий разного уровня сложности в соответствии с задачами, которые предлагались обучающимся на занятиях.

Для статистического анализа и обработки полученных данных диагностических срезов были использованы возможности интеллектуальной оболочки «Advanced Tester» [11, 12]. В этой программе можно не только получить количественный анализ экспериментальных данных, но и на его основе смоделировать индивидуальные и групповые траектории обучения. При этом отметим, что экспериментальная деятельность также предполагала и самостоятельные занятия обучающихся над учебным материалом, в том числе и в удаленном формате во внеучебное время с использованием программы [13, 14]. Для этого не требовалось вмешательства учителя, задания формировались с помощью автоматизированного подбора в соответствии с образовательными картами обучающихся.

Качественный анализ условий и результатов эксперимента

Результаты проведенного исследования отражают рост уровня знаний и сформированности умений у обучающихся во всех выделяемых категориях. При этом отчетлива видна тенденция к смещению уровня подготовки в область высоких результатов. Практически все участники педагогического эксперимента демонстрируют устойчивые навыки решения заданий повышенного и высокого уровня сложности по теме «Кодирование информации с использованием систем счисления» с применением инструментов языка Python среды программирования PyCharm.

Использование систем программирования существенно расширяет круг рассматриваемых задач. Это позволяет детально познакомить обучающихся с новыми для них идеями решения заданий анализа и вычисления количества комбинаций записей с помощью композиций алгоритмических конструкций и применения функций встроенных библиотек языка Python. Обучающиеся оттачивают мастерство написания программного кода в разнообразных сложных ситуациях, тем самым расширяя фундаментальную базу своих умений, как в области анализа записи чисел, так и в сфере приложения знаний по алгоритмизации и программированию. Кроме того, специфика программной оболочки «Advanced Tester» обеспечивает индиви-

дуальное и групповое обучение учащихся в зоне ближайшего развития [15]. Автоматизированный подбор индивидуальных заданий для обучающихся с учетом сформированности их знаний и умений и многомерный анализ усвоенных и неусвоенных компонентов учебного материала закладывает максимальный прирост уровня учебных достижений. Это позволяет утверждать, что гипотеза исследования о повышении эффективности обучения анализу и вычислению количества комбинаций записей с использованием алгоритмических конструкций и инструментов подключаемых библиотек языка программирования Python находит свое доказательство.

Заключение

Таким образом, результаты экспериментальной работы демонстрируют необходимость изучения использования инструментов языка Python среды программирования PyCharm при решении задач анализа и вычисления количества комбинаций записей чисел в разных системах счисления. Обучающиеся находят отражение изученного ранее учебного материала об основных алгоритмических конструкциях в решении задач практического вычисления комбинаторных комбинаций. При этом возможности встроенных библиотек позволяют расширить их профессиональный кругозор и найти эффективное применение знаниям об итерируемых объектах. Обучающиеся начинают «видеть» приложение теоретических подходов эффективного анализа выражений на практике. Это способствует формированию у них крепких навыков будущей профессиональной деятельности в области IT-технологий.

Список литературы

1. Тихвинский В.И., Холмогоров В.В. Системы счисления в древности и современности // International Journal of Open Information Technologies. 2022. Т. 10, № 6. С. 46–56.
2. Ветохин А.Н., Дежа Е.И., Модель Д.Л. О месте комбинаторики в математической подготовке школьников // Наука и школа. 2023. № 6. С. 160–172. DOI: 10.31862/1819-463X-2023-6-160-172.
3. Вересович С.Л. Применение комбинаторики к решению некоторых задач // Вестник Инновационного Евразийского университета. 2017. № 1 (65). С. 57–62.
4. Ильченко О.Ю., Сырицына В.Н., Кадеева О.Е. Решение задач ЕГЭ по информатике средствами языка Python // Высшее образование сегодня. 2021. № 11–12. С. 42–54. DOI: 10.18137/RNU.HET.21.11-12.P.042.
5. Ледовская Я.О., Щёкина Е.О. Современные направления применения задач комбинаторики // Научное обозрение. Педагогические науки. 2019. № 4–3. С. 64–67.
6. Козлов С.В., Быков А.А. Обучение анализу значений арифметических выражений с использованием языка программирования Python в школе // Современные наукоемкие технологии. 2024. № 5–1. С. 148–153. DOI: 10.17513/snt.40020.

7. Каган Э.М. Применение визуальных языков программирования для повышения эффективности обучения разделу «Алгоритмизация и программирование» школьного курса информатики // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2018. № 1 (43). С. 99–104.
8. Бровка Н.В., Францкевич А.А. Обучение учащихся основам алгоритмизации и программирования // Педагогика информатики. 2020. № 3. С. 1–9.
9. Пономарев Д.А. Решение олимпиадных задач по информатике по теме «Системы счисления» на языке программирования Python // Информационные технологии в образовании. 2021. № 4. С. 193–198.
10. Аванесян А.А., Разумова И.В., Донеv Д.Д. Системы счисления в математике и их применение в жизни // Университетская наука. 2022. № 1 (13). С. 131–136.
11. Козлов С.В., Быков А.А. Особенности изучения междисциплинарных тем школьных курсов математики и информатики с помощью методов математического моделирования // Проблемы современного образования. 2021. № 5. С. 250–261. DOI: 10.31862/2218-8711-2021-5-250-261.
12. Козлов С.В., Быков А.А. Применение методов математического моделирования для диагностики знаний школьников // Современные наукоемкие технологии. 2021. № 4. С. 157–162. DOI: 10.17513/snt.38632.
13. Senkina G.E., Timofeeva N.M., Kiseleva O.M. Modernization of traditional educational forms in the context of distance learning // Journal of Higher Education Theory and Practice. 2022. Т. 22. № 3. P. 160–165. DOI: 10.33423/jhetp.v22i3.5091.
14. Киселева О.М. Использование математических методов для формализации элементов образовательного процесса // Концепт. 2013. № 2. С. 51–57.
15. Козлов С.В., Быков А.А. Особенности обучения школьников олимпиадному программированию с использованием методов математического моделирования // Современные наукоемкие технологии. 2022. № 6. С. 141–146. DOI: 10.17513/snt.39215.