

УДК 004.4 + 519.6

Н.М. ЗАЩЕПКИНА, К.М. БОЖКО, Д.С. МЕЛЬНИЧЕНКО

Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"

ГРАФІЧНІ ЗАДАЧІ З КОМБІНАТОРИКИ НА ОСНОВІ ЦИФРОВОЇ КЛАВІАТУРИ

Мобільні додатки у наш час є предметом бурхливого розвитку та дослідження. Для розробки мобільних додатків використовують класичні алгоритми. Проте, підготовка програмістів для мобільних пристроїв потребує більшого використання прикладів і задач із графічним наповненням.

Запропоновано, як колись це сталося із шахівницею, використати цифрову клавіатуру телефону для створення та розв'язання задач комбінаторики. В статті сформульовано задачу на розміщення: розбити клавіатуру мобільного телефону на три групи по три клавіші таким чином, щоб вони утворили трійку простих чисел. Цифру 0 не використовувати.

Дана задача є наочною і може бути вирішена із застосуванням аналітичних та графічних засобів. Наочні задачі комбінаторики посідають чільне місце в збірках задач з дискретної математики, теорії алгоритмів, програмування тощо. Графічні або графоаналітичні методи розв'язання задач комбінаторики можна зустріти на багатьох прикладах із шахівницею. Поширеними задачами комбінаторики є пошук різних комбінацій із простих чисел. Прості числа знаходять за допомогою алгоритму Ератосфена або його похідних.

Запропоновані задачі пошуку простих чисел та їх комбінацій із додаванням графічного інструментарію, в якості якого є цифрова клавіатура телефону або іншого побутового пристрою. Загальний алгоритм розв'язання задачі розбиття клавіатури телефону на три групи по три цифри (без нуля) так, щоб утворити три простих числа. Було отримано додаткову умову для обмеження пошуку: не може бути в знайденому числі трьох цифр із набору.

Розв'язано задачу пошуку розміщення по три цифри для дев'яти цифр клавіатури із максимальною сумою: $947 + 821 + 653 = 2421$. Мовою Java був створений мобільний додаток із розв'язанням задачі розбиття клавіатури телефону на три простих числа. Задачі комбінаторики із графічною складовою є наочними і сприятимуть швидкому опануванню алгоритмів розв'язання задач програмування та вивчення мов, зокрема такої складної для початківців, якою є мова Java. Програма наведена в статті. Запропоновано додати до наочних інструментів комбінаторики цифрову клавіатуру, яка є звичним елементом побуту сучасної людини.

Ключові слова: алгоритм Ератосфена, прості числа, мобільний додаток.

Н.М. ЗАЩЕПКИНА, К.М. БОЖКО, Д.С. МЕЛЬНИЧЕНКО

Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"

ГРАФИЧЕСКИЕ ЗАДАЧИ ПО КОМБИНАТОРИКИ НА ОСНОВЕ ЦИФРОВОЙ КЛАВИАТУРЫ

Мобильные приложения в наше время является предметом бурного развития и исследования. Для разработки мобильных приложений используют классические алгоритмы. Однако, подготовка программистов для мобильных устройств требует большего использования примеров и задач с графическим наполнением.

Предложено, как когда-то это случилось с шахматной доской, использовать цифровую клавиатуру для создания и решения задач комбинаторики. В статье сформулирована задача на размещение: разбить клавиатуру мобильного телефона на три группы по три клавиши таким образом, чтобы они образовали тройку простых чисел. Цифру 0 не использовать.

Данная задача является наглядной и может быть решена с применением аналитических и графических средств. Наглядные задачи комбинаторики занимают видное место в сборниках задач по дискретной математике, теории алгоритмов, программирование и тому подобное. Графические или графоаналитические методы решения задач комбинаторики можно встретить на многих примерах с шахматной доской. Распространенными задачами комбинаторики является поиск различных комбинаций из простых чисел. Простые числа находят с помощью алгоритма Эратосфена или его производных. Предложенные задачи поиска простых чисел и их комбинаций с добавлением графического инструментария, в качестве которого является цифровая клавиатура телефона или другого бытового устройства. Общий алгоритм решения задачи разбиения клавиатуры телефона на три группы по три цифры (без нуля) так, чтобы образовать три простых числа. Было получено дополнительное условие для

ограничения поиска: не может быть в найденном числе трех цифр из набора. Решена задача поиска размещения по три цифры для девяти цифр клавиатуры с максимальной суммой: $947 + 821 + 653 = 2421$.

На языке Java был создан мобильное приложение с решением задачи разбиения клавиатуры телефона на три простых числа. Задачи комбинаторики с графической составляющей является наглядными и способствуют быстрому овладению алгоритмов решения задач программирования и изучение языков, в частности такой сложной для начинающих. Программа приведена в статье. Предложено добавить в наглядные инструменты комбинаторики цифровую клавиатуру, которая является обычным элементом быта современного человека.

Ключевые слова: алгоритм Эратосфена, простые числа, мобильное приложение.

N.M. ZASHCHERKINA, K.M. BOZHKO, D.S. MELNICHENKO
National Technical University of Ukraine
"Ihor Sikorsky Kiev Polytechnic Institute"

GRAPHIC PROBLEMS FROM COMBINATORS ON THE BASIS OF DIGITAL KEYBOARD

Mobile additions in our time are the article of rapid development and research. For development of mobile additions use classic algorithms. However, preparation of programmers for mobile devices need the greater use of examples and tasks with the graphic filling. It offers, as once it happened to chess-board, to use the digital keyboard of telephone for creation and decision of tasks of combinatorics.

In the article a task is set forth on placing: to divide the keyboard of mobile telephone into three groups for three keys thus, that they formed three of prime numbers. Not to use a number 0.

This task is evident and can be decided with application of analytical and graphic facilities. The evident tasks of combinatorics occupy a main place in collections of tasks from discrete mathematics, theories of algorithms, programming and others like that. The graphic or graphico-analytical methods of decision of tasks of combinatorics can be met on many examples with chess-board.

The widespread tasks of combinatorics is a search of different combinations from prime numbers. Prime numbers find by means of algorithm of Eratosfen or his derivatives. Offer tasks of search of prime numbers and their combinations are with addition of graphic tool, as that there is a digital keyboard of telephone or other domestic device.

A general algorithm of decision of task of laying out of keyboard of telephone is on three groups for three numbers (without a zero) so that to form three prime numbers. An additional condition was got for limitation of search : it can not be in the found number of three numbers from a set. The task of search of placing is decided for three numbers for nine numbers of keyboard with a maximal sum: $947 + 821 + 653 = 2421$.

By the language of Java mobile addition was created with the decision of task of laying out of keyboard of telephone on three prime numbers. Tasks of combinatorics with a graphic constituent are evident and will assist the rapid capture of algorithms of decision of tasks of programming and study of languages, in particular such difficult for beginners.

The program is presented in the article. It offers to add to the evident instruments of combinatorics a digital keyboard that is the usual element of way of life of modern man.

Keywords: Eratosfenes algorithm, simple numbers, mobile application.

Постановка проблеми

Наочні задачі комбінаторики посідають чільне місце в збірках задач з дискретної математики, теорії алгоритмів, програмування тощо. Графічні або графоаналітичні методи вирішення задач комбінаторики можна зустріти на багатьох прикладах із шахівниці [1].

Мобільні додатки у наш час є предметом бурхливого розвитку та дослідження. Для розробки мобільних додатків використовують класичні алгоритми. Проте, підготовка програмістів для мобільних пристроїв потребує, на нашу думку, більшого використання прикладів і задач із графічним наповненням. Нами запропоновано, як колись це сталося із шахівницею, використати цифрову клавіатуру телефону для створення та розв'язання задач комбінаторики.

Аналіз останніх досліджень і публікацій

Поширеними задачами комбінаторики є пошук різних комбінацій із простих чисел. Прості числа знаходять за допомогою алгоритму Ератосфена або його похідних. Так, для знаходження всіх простих чисел, які не перевищують задане число n , відповідно до методу Ератосфена необхідно виконати наступні кроки:

1. Виписати підряд всі цілі числа від 2 до n .
2. Нехай змінна p спочатку дорівнює двом – першому простому числу.
3. Закреслити у списку числа від $2p$ до n . При цьому рахують кроками по p (це будуть числа, які є кратними до p : $2p, 3p, 4p, \dots$).
4. Знайти перше незакреслене число у списку, яке є більшим за p .

5. Повторювати кроки 3 та 4 доки змінна p не досягне значення n .
 Результат: усі незакреслені числа від 2 до n є простими.
 Алгоритм Ератосфена реалізовано на С++ та іншими мовами програмування [2, 3].
 Алгоритм Ератосфена досліджений на складність [4], яку для максимального цілого числа n визначають за формулою:

$$S = (n \ln(\ln n)). \tag{1}$$

У монографіях та підручниках з теорії чисел [5] можна знайти точний доказ формули (1).
 Великі прості числа використовують у алгоритмі роботи криптографічної системи із відкритим ключем RSA [6], що надає задачам із простими числами практичного інтересу.

Мета дослідження

Розв’язати задачу на розміщення: розбити клавіатуру мобільного телефону на три групи по три клавіші таким чином, щоб вони утворили трійку простих чисел. Цифру 0 не використовувати. Одне із рішень зображене на рис. 1. Дана задача є наочною і може бути вирішена із застосуванням аналітичних та графічних засобів.

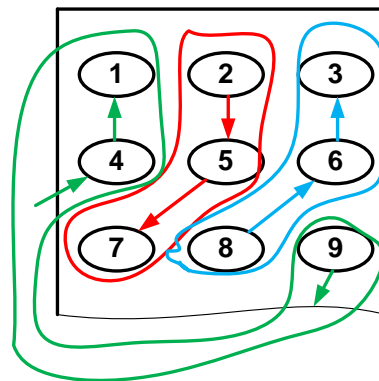


Рис. 1. Розбиття цифрової клавіатури на групи, які утворюють три простих числа: червона зона – число 257, синя зона – число 863, зелена зона – число 941

Викладення основного матеріалу дослідження

Задачу на розміщення формулюють таким чином: розбити множину із n різних елементів на групи по k різних елементів в кожній. При цьому вважають розбиття новим, якщо при однакових елементах в групі порядок їх слідування є іншим. Кількість варіантів розміщення визначають за формулою:

$$A(n, k) = n!/(n-k)! \tag{2}$$

Для 9 цифр і 3 груп маємо: $A = 504$. Таким чином, із 9 цифр можна утворити 504 комбінації по три числа із різними цифрами.

При розв’язанні поставленої задачі, а саме знайти прості числа, основним є алгоритм Ератосфена. Загальний алгоритм розв’язання задачі розбиття клавіатури телефону на три групи по три цифри (без нуля) так, щоб утворити три простих числа сформулюємо наступним чином:

1. Знайти тризначні числа, які є простими.
2. Знайдені числа не мають цифри "0".
3. Усі цифри в кожному із знайдених чисел є різними.
4. Нами отримано також додаткову умову для обмеження пошуку: не може бути в знайденому числі трьох цифр із набору {1, 3, 7, 9}.

Усього було знайдено 67 чисел, які задовольняють вимогам 1–4. Проте, тільки 59 чисел утворюють розміщення із трьох чисел (групи по три числа із різних цифр). В табл. 1 наведені ці числа та їх кількість по сотнях.

Таблиця 1

Прості тризначні числа та їх кількість по сотнях									
Прості числа	149	241, 251, 257, 263, 269, 281, 283, 293	347, 359, 367, 389	419, 421, 431, 457, 461, 463, 467, 479, 487, 491	521, 523, 541, 563, 569, 571, 587, 593	613, 619, 631, 641, 643, 647, 653, 659, 673, 683	743, 751, 769	821, 823, 827, 829, 839, 853, 857, 859, 863, 893	941, 947, 953, 967, 983
Усього	1	8	4	10	8	10	3	10	5

Вирішено також задачу пошуку розміщення по три цифри для дев'яти цифр клавіатури із максимальною сумою: $947 + 821 + 653 = 2421$.

Мовою Java був створений мобільний додаток із вирішенням задачі розбиття клавіатури телефону на три простих числа. Текст програми наведено нижче.

```
//Класс для проверки цифр в трехзначных числах на повторение
public class PrimeNumberCombination {
public void Combination(int One, int Two, int Three)
{//Три числа которые нам передал главный метод
boolean good = true; //Переменная для подтверждения удачной комбинации //
int ComboNumber[] = new int [9]; //Создаем массив длиной в 9 чисел
ComboNumber[0] = One%10; //Последняя цифра первого числа
ComboNumber[1] = (One/10)%10; //Вторая цифра первого числа
ComboNumber[2] = One/100; //Первая цифра первого числа
ComboNumber[3] = Two%10; //Первая цифра второго числа
ComboNumber[4] = (Two/10)%10; //Первая цифра второго числа
ComboNumber[5] = Two/100; //Первая цифра второго числа
ComboNumber[6] = Three%10; //Первая цифра третьего числа
ComboNumber[7] = (Three/10)%10; //Первая цифра третьего числа
ComboNumber[8] = Three/100; //Первая цифра третьего числа

//Будем сравнивать цифры с 0 по 9 потом с 1 по 9 и т.д
for(int sravn = 0; sravn < 9; sravn++) {
for(int sravnTwo = sravn+1; sravnTwo < 9; sravnTwo++) {
//Если хоть одна цифра совпадет в двух числах, то good поменяется с
True на false
if(ComboNumber[sravn] == ComboNumber[sravnTwo]) {
good = false;
break; //Выйти из цикла, который sravn+1
}
}
if(good == false) {
break; //Если good = false, то выйти из цикла того, что sravn
}
}
// если после операции сравнения good не поменялось на false, то мы
выведем все 3 числа
if(good == true) {
System.out.println(" "+One+" "+Two+" "+Three);
}
}
}
PrimeNumberCombination.java
/*Класс для нахождения простых чисел
методом перебора делителей до корня из числа*/
public class PrimeNumberFound {
public int found(int number){ //В метод приходит число из главного класса
double squerNumber = Math.sqrt(number); //Ищем корень полученного числа
int sqrtNumber = (int) squerNumber; //Убираем все после точки из корня
//Начиная с двойки перебираем все числа с 2 до корня и если на хоть одно
делиться без остатка, то оно нам не подходит
for (int point = 2; point < sqrtNumber; point++) { //Шаг 3
if(number%point==0) {
return 0;
}
}
}
//Если Шаг 3 прошел и не вернул ноль, то значит число простое и вернем
его в качестве результата работы
return number;
}
}
```

```

PrimeNumberFound.java
    public class TestMainClass {

        public static void main(String[] args) {
            int FirstNumber;

int LastNumber=999;//Последнее число, до которого мы дойдем во время проверки
int primeMass [];//Место для простых чисел
    primeMass = new int[1000];
    PrimeNumberFound OnceNumber = new PrimeNumberFound();//Создаем экземпляр
из класса для простых чисел
    for(FirstNumber = 100; FirstNumber<LastNumber; FirstNumber++)
    { //(Шаг 1)Цикл с первого числа до последнего
int primeNumber = OnceNumber.found(FirstNumber);//вызываем метод found
if(primeNumber != 0) { //Если метод возвращает не ноль,
                        то записать возвращенное число в массив
    primeMass[FirstNumber] = primeNumber;//Запись в массив
        }
    }
int length=primeMass.length;//Узнаем длину полученного массива простых чисел
//Данный цикл нужен, чтобы убрать простые числа с 0 (Шаг 2)
for(int Zero = 0; Zero<length; Zero++) {
    if(((primeMass[Zero]/10)%10) == 0) {
        primeMass[Zero] = 0;
    }
}
//поскольку после Шаг 1 и Шаг 2 в массиве полно нулей,
его нужно отсортировать, чтобы нули были в конце
    for(int serialO = 0; serialO<length; serialO++) {
        for(int serialT = 0; serialT<length; serialT++) {
            int timel = 0;
            if(primeMass[serialT]<primeMass[serialO]) {
                timel = primeMass[serialO];
                primeMass[serialO] = primeMass[serialT];
                primeMass[serialT] = timel;
            }
        }
    }
//Дальше нужно обрезать уже отсортированный массив по последнему нулю,
для этого узнаем номер в массиве первого нуля
    int primeMassLength = 0;//Запомним номер первого нуля
    for(int cut = 0; cut<length; cut++) {
        if(primeMass[cut] == 0) { //как только число в массиве равно
нулю, записать его и прервать цикл
            primeMassLength = cut;
            break;
        }
    }
//Теперь будем по очереди выбирать числа из массива для комбинации
for(int pointOne = 0; pointOne < primeMassLength; pointOne++ ) {
    int One = primeMass[pointOne]; //Запомнить первое выбранное число
for(int pointTwo = 0; pointTwo < primeMassLength; pointTwo++) {
    int Two = primeMass[pointTwo]; //Второе выбранное число
for(int pointThree = 0; pointThree< primeMassLength; pointThree++ ){
    int Three = primeMass[pointThree]; //Третье выбранное число
PrimeNumberCombination Don = new PrimeNumberCombination();
//Создаем экземпляр класса для проверки комбинаций из трех выбранных чисел
Don.Combination(One, Two, Three);
//Передаем 3 выбранных числа в гл. метод
        }
    }
}
}

```

```

    }
}
}
TestMainClass.java

```

Результат роботи програми виведено у вигляді таблиці на екран. Фрагмент розв'язку наведено на рис. 2. Розв'язок – це трійки чисел, що задовольняють вимогам 1-4, які були визначені вище.

```

<terminated> TestMainClass [Java Application] D:\Java\bin\javaw.exe (26 февр. 2018 г., 8:36:40)
983 647 521
983 647 251
983 641 257
983 521 647
983 521 467
983 467 521
983 467 251
983 461 257
983 257 641
983 257 461
983 251 647
983 251 467
967 853 421
967 853 241
967 841 523
967 823 541
967 541 823
967 541 283
967 523 841
967 421 853
967 283 541
967 241 853
961 823 547
961 823 457
961 547 823
961 547 283
961 523 487
961 487 523
961 457 823
961 457 283
961 283 547
961 283 457
953 827 641
953 827 461
953 821 647
953 821 467
953 647 821
953 647 281
953 641 827
953 467 821
953 467 281
953 461 827

```

Рис. 2. Результат роботи мобільного додатку: розміщення по три різні цифри із 9 цифр клавіатури, які утворюють тризначні прості числа

Висновки

Задачі комбінаторики із графічною складовою є наочними і сприятимуть швидкому опануванню алгоритмів вирішення задач програмування та вивчення мов, зокрема такої складної для початківців. Якою є мова Java. Нами запропоновано додати до наочних інструментів комбінаторики цифрову клавіатуру, яка є звичним елементом побуту сучасної людини.

Список використаної літератури

1. Виленкин Н.Я. Популярная комбинаторика [Текст] / Н.Я. Виленкин – М. :Наука, 1975. – 208 с.
2. Седжвик Р. Алгоритмы на C++ . 2-е изд. [Текст] / Р. Седжвик. – М. : Интуит, 2016. – 1000 с.
3. Sorenson Jonathan. An Introduction to Prime Number Sieves [Текст] / Jonathan Sorenson // Computer Sciences Technical Report #909, Department of Computer Sciences University of Wisconsin-Madison. – January 2 1990. – 15 p.
4. Pritchard Paul. Linear prime-number sieves: a family tree [Текст] / Р. Pritchard // Sci. Comput. Programming. – 1987. – № 9:1. – P. 17-35.
5. Hardy G.H. An Introduction to the Theory of Numbers 4-ed. [Текст] / G.H. Hardy, E.M. Wright – Oxford. : Clarendon Press? 1975. – 421 p.
6. Кормен Т. Алгоритмы: построение и анализ. 2-е изд. [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – К.: Вильямс, 2012. – 1296 с.