МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ: ЗАДАЧА О РАЗБИЕНИИ НА РАВНЫЕ ЧАСТИ

ЛАБОРАТОРНАЯ РАБОТА

студента 3 курса 331 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Токарева Никиты Сергеевича

Проверил
доцент _____ А. Н. Гамова

Саратов 2022

СОДЕРЖАНИЕ

1 Разбиение множества на равные части

Необходимо выполнить следующие задачи:

1. Вычислить сумму sum массива. Если сумма нечетная, не может быть двух подмножеств с одинаковой суммой, поэтому верните false;
2. Если сумма элементов массива четная, вычислить $sum_{\frac{1}{2}} = sum/2$ найти подмножество массива с суммой, равной $sum_{\frac{1}{2}}$.
   Вычисление подмножества происходит следующим образом:
   Вход: Конечное множество символов S мощности n и сумма $sum_{\frac{1}{2}}$.

   Выход: Подмножества A и $\overline{A}$ такие что $S = A \cup \overline{A}$.

1. Инициализировать переменную $cnt = sum_{\frac{1}{2}}$, создать список $list = [x_0, \ldots, x_{n-1}]$, где $x_i \in S$ $(1 \leq i < n)$ и пустой список $partition = []$.
2. Пока переменная $cnt! = 0$ выполнять: $cnt = cnt - list[n-1]$, добавить в $partition$ элемент $list[n-1]$, $n = n - 1$.
3. Может оказаться так, что cnt станет меньше нуля. В таком случае вернуться к первому пункту. Если $cnt = 0$, то мы получили подмножество, элементы которого находятся в $partition$.
4. Далее найдем элементы множества $S \backslash partition$.

2 Код программы, реализующий алгоритм

Далее представлена реализация данного алгоритма, написанная на языке *Python*.

```
import numpy as np

def isSubsetSum(set_list, n, sum):
        if sum == 0:
                return True
        if n == 0 and sum != 0:
                return False

        if set_list[n - 1] > sum:
                return isSubsetSum(set_list, n - 1, sum)

        return isSubsetSum(set_list, n - 1, sum) \
        or isSubsetSum(set_list, n - 1, sum - set_list[n - 1])


def findPartition(set_list, n, sum):
        if sum % 2 != 0:
                return False

        return isSubsetSum(set_list, n, sum // 2)


def print_set(s, n):
    print('{', end=' ')
    k = 1
    for el in s:
        if k == n:
            print(el, '}', end=' ')
        else:
            print(str(el) + ',', end=' ')
        k += 1


def divide_set(set_list, sum):
        cnt = sum
        n = len(set_list)
        tmp_list = set_list
```

```
        part = []
        while cnt != 0:
                if cnt < 0:
                        tmp_list = np.random.permutation(set_list)
                        cnt = sum
                        n = len(set_list)
                        part = []
                cnt -= tmp_list[n - 1]
                part.append(tmp_list[n - 1])
                n -= 1
        part.sort()
        scnd_part = []
        for el in set_list:
                if el not in part:
                        scnd_part.append(el)
        return part, scnd_part


print('Введите через пробел элементы множества:')
s = input()
set_list = [int(i) for i in s.split(' ')]
set_list = np.array(set_list)

sum = 0
for el in set_list:
        sum += el

if findPartition(set_list, len(set_list), sum) == True:
        print("Данное множество можно разделить на два подмножества равной суммы")
        frst, scnd = divide_set(set_list, sum // 2)
        print('Например:', end=' ')
        print_set(frst, len(frst))
        print_set(scnd, len(scnd))

else:
        print("Данное множество нельзя разделить на два подмножества равной суммы")
```

На рисунке 1 представлен результат работы алгоритма.

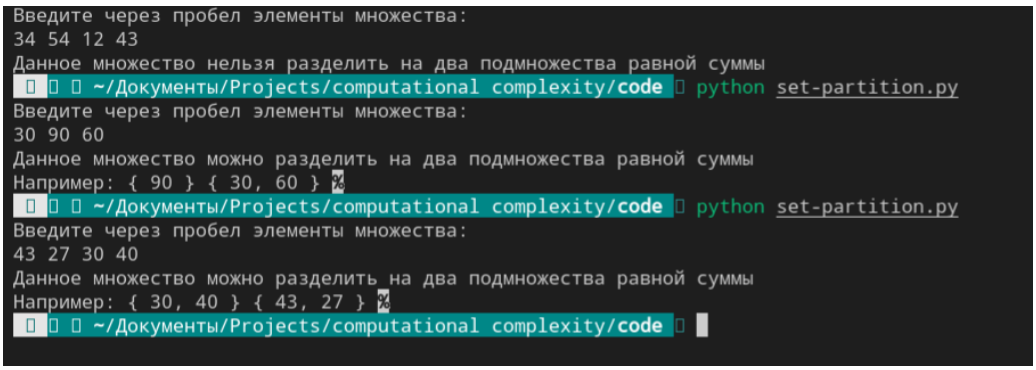

Рисунок 1 – Результат работы алгоритма

Анализ алгоритма

Очевидно, что сложность данного алгоритма равна $O(2^n)$, так как в худшем случае это решение пробует две возможности (включить или исключить текущий элемент) для каждого элемента.

ЗАКЛЮЧЕНИЕ

В данной лабораторной работе был рассмотрен алгоритм разбиения множества на подможества с одинаковыми суммами и был проведен анализ оценки его сложности. Это послужило созданием её программной реализации, написанной на языке $Python$.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Статья «Partition of the set» / [Электронный ресурс] URL: https://www.tutorialspoint.com/partitioning-of-a-set (дата обращения 02.05.2022), Яз. рус.

2 Книга Томаса Кормена «Алгоритмы» / [Электронный ресурс] URL: https://e-maxx.ru/bookz/files/cormen.pdf (дата обращения 02.05.2022), Яз. рус.