

Университет ИТМО

Инфокоммуникационные технологии и  
системы связи

Программирование в  
инфокоммуникационных сетях

Лабораторная работа #1

Коркина Анна Михайловна (К3120)

Преподаватель: Харьковская Татьяна  
Александровна

Г. Санкт-Петербург

10.04.2021

## 1 задача. Максимальная стоимость добычи

Вор находит гораздо больше добычи, чем может поместиться в его сумке. Помогите ему найти самую ценную комбинацию предметов, предполагая, что любая часть предмета добычи может быть помещена в его сумку.

**Решение:** сортируем массив цен и весов по цене за килограмм по убыванию и забираем всё пока не закончится место в рюкзаке.

**Код:**

```
in_file = open('input.txt')
n, max_weight = in_file.readline().split()
n, max_weight = int(n), int(max_weight)
things = []
for i in range(int(n)):
    line = in_file.readline().split()
    price = int(line[0])
    weight = int(line[1])
    things.append((price, weight, price / weight))
in_file.close()

things.sort(key=lambda x: x[2], reverse=True)
taked = 0

ans = ""
for i in things:
    new_weight = max_weight - i[1]
    if (new_weight >= 0):
        taked += i[0]
        max_weight -= i[1]
    else:
        taked += i[0] * (max_weight / i[1])
        max_weight = 0

    if (max_weight == 0):
        ans = f"{taked:.4f}"

out = open('output.txt', 'w')
out.write(ans)
print(ans)
out.close()
```

## 2 задача. Заправки

Вы собираетесь поехать в другой город, расположенный в  $d$  км от вашего родного города. Ваш автомобиль может проехать не более  $m$  км на полном баке, и

вы начинаете с полным баком. По пути есть заправочные станции на расстояниях

$stop_1, stop_2, \dots, stop(n)$  из вашего родного города. Какое минимальное количество

заправок необходимо?

**Решение:**

В данном решении осуществляется проход по заправкам и смотрится возможно ли доехать до следующей заправки без пополнения бака на текущей.

**Код:**

```
inp = open('input.txt')
need = int(inp.readline())
tank = int(inp.readline())
n = int(inp.readline())
stations = list(map(int, inp.readline().split()))
inp.close()
```

```
def getFillsCount(need, tank, stations):
    stations.append(need)
    curr = 0
    count = 0
    for i in range(len(stations) - 1):
        if (stations[i] - curr > tank):
            return -1
        if (stations[i + 1] - curr) > tank:
            curr = stations[i]
            count += 1
    if (stations[len(stations) - 1] - curr < tank):
        return count
    else:
        return -1
```

```
count = getFillsCount(need, tank, stations)
out = open('output.txt', 'w')
out.write(str(count))
print(str(count))
out.close()
```

## 3 задача. Максимальный доход от рекламы

У вас есть  $n$  объявлений для размещения на популярной интернет-странице. Для каждого объявления вы знаете, сколько рекламодатель готов платить за один клик по этому объявлению. Вы настроили  $n$  слотов на своей странице и оценили ожидаемое количество кликов в день для каждого слота. Теперь ваша цель - распределить рекламу по слотам, чтобы максимизировать общий доход.

Код:

```
inp = open("input.txt")
n = int(inp.readline())
a = list(map(int, inp.readline().split()))
b = list(map(int, inp.readline().split()))
inp.close()
```

```
a = list(sorted(a))
b = list(sorted(b))
```

```
summ = 0
for i in range(n):
    summ += a[i] * b[i]
```

```
out = open("output.txt", "w")
out.write(str(summ))
out.close()
```

#### **Решение:**

изначально сортируются два входных массива и произведения соответствующих элементов из двух массивов суммируются. Это позволяет нам получить максимальный доход даже при наличии отрицательных чисел в массивах.

## 5 задача.Максимальное количество призов

Вы организуете веселый конкурс для детей. В качестве призового фонда у вас есть  $n$  конфет. Вы хотели бы использовать эти конфеты для раздачи  $k$  лучшим местам в конкурсе с естественным ограничением, заключающимся в том, что чем выше место, тем больше конфет. Чтобы осчастливить как можно больше детей, вам нужно найти наибольшее значение  $k$ , для которого это возможно.

Решение:

```
fin = open('input.txt')
n = int(fin.readline())
sum = 0
list_num = []
for i in range(1, n+1):
    if sum + i <= n:
        sum += i
        list_num.append(i)
    else:
        list_num[-1] += (n-sum)
        break
```

```
fout = open('output.txt', 'r')
fout.write(len(list_num))
fout.write(*list_num)
```

в массив для хранения потенциально возможного разложения числа помещаются все целые числа, начиная с единицы до того момента, пока их сумма не станет больше, чем разность параметра и последнего элемента, добавленного в массив. После этого в массив добавляется последнее число, необходимое для того, чтобы сделать сумму равной параметру(завершить разложение)

## 6 задача.Максимальная зарплата

В качестве последнего вопроса успешного собеседования ваш начальник дает вам несколько листов бумаги с цифрами и просит составить из этих цифр наибольшее число.

Решение:

если числа одного порядка — то проблем нет, выбираем наибольшее

иначе станное условие: если наименьшее равно концу наибольшего и больше чем его начало то оно идет на первое место (например, 233 и 33 : **33** = 2 **33**)

и **33** > **23** 3 → 33 на первое место)

## 12 задача. Последовательность

Дана последовательность натуральных чисел  $a_1, a_2, \dots, a_n$ , и известно, что  $a_i \leq i$  для любого  $1 \leq i \leq n$ . Требуется определить, можно ли разбить элементы последовательности на две части таким образом, что сумма элементов в каждой из частей будет равна половине суммы всех элементов последовательности.

**Решение:**

```
inp = open("input.txt")
n = int(inp.readline())
seq = list(map(int, inp.readline().split()))
need = sum(seq)
if(need % 2 == 1):
    out = open("output.txt", "w")
    out.write("-1")
    out.close()
    exit()

need = need // 2
taken = 0
i = 0
while((i < len(seq)) and (taken < need)):
    taken += seq[i]
    i += 1

if(taken == need):
    res = ""
    takenSecondHalf = 0
    takenSecondCount = 0
    for index in range(i, len(seq)):
        takenSecondHalf += seq[index]
        takenSecondCount += 1
        res += f"{seq[index]} "
    if (takenSecondHalf == taken == need):
        out = open("output.txt", "w")
        out.write(f"{takenSecondCount} \n")
        out.write(res)
        out.close()
        exit()

out = open("output.txt", "w")
out.write("-1")
out.close()
```

Так как разбить последовательность надо на 2 части (то есть нельзя вырывать элементы из разных частей, а надо получить две непрерывные последовательности), то сначала просто в цикле while набираем необходимую нам сумму (сумму/2), если она набралась, то суммируем оставшиеся элементы и проверяем равны ли у нас первая и вторая часть.

## 13. Сувениры

Вы и двое ваших друзей только что вернулись домой после посещения разных стран. Теперь вы хотели бы поровну разделить все сувениры, которые все трое купили.

```
def canDivision(coins, n, first, second, third, backs):
    if (first == 0) and (second == 0) and (third == 0):
        if (n < 0):
            return True
        else:
            return False

    if (n < 0):
        return False

    way = (first, second, third, n)
    if (way not in backs):
        inFirst = False
        inSecond = False
        inThird = False

        if (first - coins[n] >= 0):
            inFirst = canDivision(coins, n - 1, first - coins[n], second, third, backs)
        if (inFirst == False) and (second - coins[n] >= 0):
            inSecond = canDivision(coins, n - 1, first, second - coins[n], third,
backs)
        if (inFirst == False) and (inSecond == False) and (third - coins[n] >=
0):
            inThird = canDivision(coins, n - 1, first, second, third - coins[n],
backs)

        backs[way] = inFirst or inSecond or inThird

    return backs[way]

inp = open("input.txt")
n = int(inp.readline())
```

```

coins = list(map(int, inp.readline().split()))

summ = sum(coins)
if (summ % 3 != 0) or (n < 3):
    out = open("output.txt", "w")
    out.write("0")
    out.close()
    exit

need = summ / 3

backs = {}

res = canDivision(coins, n - 1, need, need, need, backs)

out = open("output.txt", "w")
if (res):
    out.write("1")
else:
    out.write("0")
out.close()

```

**Решение:** сначала находим сумму каждой группы, дальше нужно равномерно распределить наибольшие числа по группам, поэтому сортируем массив по убыванию и жадным алгоритмом добавляем максимальные числа в группы

## 20 задача. Почти палиндром (3 балла)

**Задача:**

Слово называется палиндромом, если его первая буква совпадает с последней, вторая – с предпоследней и т.д. Например: «abba», «madam», «x». Для заданного числа K слово называется почти палиндромом, если в нем можно изменить не более K любых букв так, чтобы получился палиндром).

**Решение:**

Изначально перебираются все под слова которые можно составить из данного слова, и для каждого вызывается функция которая проверяет сколько “ошибок палиндрома” оно содержит, и если это количество не превосходит K, то мы засчитываем этот вариант. И в конце проверяется является ли само слово почти палиндромом.

Код:



```

inp = open('input.txt')
N, K = map(int, inp.readline().split())
word = inp.readline()
inp.close()

def check_palidrom(word, K):
    if len(word) == 1:
        return True
    first_half = word[:len(word) // 2]
    if len(word) % 2 == 0:
        second_half = word[len(word) // 2:]
    else:
        second_half = word[len(word) // 2 + 1:]
    errorsCount = 0
    for i in range(len(first_half)):
        if first_half[i] != second_half[len(second_half) - i - 1]:
            errorsCount += 1
        if errorsCount > K:
            return False
    return True

count = 0
for size in range(1, len(word)):
    for i in range(len(word) - size + 1):
        sub_word = word[i:i + size]
        if check_palidrom(sub_word, K):
            count += 1

if (check_palidrom(word, K)):
    count += 1

out = open('output.txt', 'w')
out.write(str(count))
out.close()

```

Задача не прошла ограничение на время на 11 тесте:

Скриншот веб-интерфейса с результатами тестов. Таблица содержит следующие данные:

ID	Дата	Автор	Задача	Язык	Результат	Тест	Время	Память
17088876	04.05.2022 11:16:58	Кравченко Алексей Андреевич	0576	Python	Wrong answer	1		362 KБ
17088875	04.05.2022 11:16:47	Цой Максим Андреевич	0983	C++	Accepted		0,218	3540 KБ
17088874	04.05.2022 11:16:39	Кравченко Алексей Андреевич	0576	Python	Wrong answer	4	0,015	358 KБ
17088873	04.05.2022 11:16:24	Корова Анна Михайловна	0268	Python	Time limit exceeded	11	1,234	510 KБ
17088872	04.05.2022 11:16:19	Петтиратов А.В.	0297	Python	Accepted		0,046	356 KБ
17088871	04.05.2022 11:15:54	Кузнецов Николай Сергеевич	0268	PyPy	Wrong answer	1		524 KБ
17088870	04.05.2022 11:15:44	Садыхов Эльмир Русланович	0002	Python	Wrong answer	3	0,031	366 KБ
17088869	04.05.2022 11:15:36	Белов Александр	0112	C++	Time limit exceeded	10	0,812	672 KБ
17088868	04.05.2022 11:15:29	Петтиратов А.В.	0297	Python	Wrong answer	8	0,031	370 KБ
17088867	04.05.2022 11:15:28	Зубово Михаил Петрович	0756	Python	Wrong answer	3	0,031	366 KБ
17088866	04.05.2022 11:15:10	Кузнецов Николай Сергеевич	0268	PyPy	Time limit exceeded	11	1,187	4596 KБ
17088865	04.05.2022 11:15:08	Сорока Артем Сергеевич	0007	C++	Wrong answer	2	0,015	412 KБ
17088864	04.05.2022 11:15:00	Тынышбеков	0002	Java	Wrong answer	3	0,04	1204 KБ
17088863	04.05.2022 11:14:56	Салихов Булат Марсеевич	0131	C++	Accepted		0,015	424 KБ
17088862	04.05.2022 11:14:52	Трусов Иван	0833	Python	Accepted		0,046	362 KБ
17088861	04.05.2022 11:14:32	Цой Максим Андреевич	0983	C++	Wrong answer	7	0,156	1772 KБ
17088860	04.05.2022 11:14:29	Трусов Иван	1109	Python	Accepted		0,031	362 KБ
17088859	04.05.2022 11:14:21	Кравченко Алексей Андреевич	0894	Python	Accepted		0,062	1796 KБ
17088858	04.05.2022 11:13:57	Турсалбек Ажаал	0006	Visual C++	Wrong answer	1		288 KБ
17088857	04.05.2022 11:13:49	Каширина Оксана Сергеевна	1127	C++	Accepted		0,03	420 KБ

Внизу экрана виден баннер Яндекс Директ с предложением получить промокод на 5000 рублей.

## 21 задача. Игра в дурака (3 балла)

Петя очень любит программировать. Недавно он решил реализовать популярную карточную игру «Дурак». Но у Пети пока маловато опыта, ему срочно нужна Ваша помощь. Как известно, в «Дурака» играют колодой из 36 карт. В Петиной программе каждая карта представляется в виде строки из двух символов, где первый символ означает ранг ('6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A') карты, а второй символ означает масть ('S', 'C', 'D', 'H'). Ранги перечислены в порядке возрастания старшинства.

**Решение:**

```
def getCardWeight(card):
    arr = ["6", "7", "8", "9", "T", "J", "Q", "K", "A"]
    return arr.index(card)

inp = open("input.txt")

myCount, enemyCount, trump = inp.readline().split()
myCount = int(myCount)
enemyCount = int(enemyCount)
myCards = inp.readline().split()
enemyCards = inp.readline().split()

myColorCards = {"S": [], "C": [], "D": [], "H": []}
enemyColorCards = {"S": [], "C": [], "D": [], "H": []}

for card in myCards:
    myColorCards[card[1]].append(card[0])
for card in enemyCards:
    enemyColorCards[card[1]].append(card[0])

cardColors = ["S", "C", "D", "H"]
for color in cardColors:
    myColorCards[color] = sorted(myColorCards[color], key=lambda x:
getCardWeight(x))
    enemyColorCards[color] = sorted(enemyColorCards[color], key=lambda x:
getCardWeight(x), reverse=True)
cardColors.remove(trump)

def CanWin():
    for enemyCard in enemyColorCards[trump]:
        isBroke = False
        for myCard in myColorCards[trump]:
            if (getCardWeight(myCard) > getCardWeight(enemyCard)):
```

```

        myColorCards[trump].remove(myCard)
        isBroke = True
        break
    if (isBroke == False):
        return False

    for color in cardColors:
        for enemyCard in enemyColorCards[color]:
            isBroke = False
            for myCard in myColorCards[color]:
                if (getCardWeight(myCard) > getCardWeight(enemyCard)):
                    myColorCards[color].remove(myCard)
                    isBroke = True
                    break
            if (isBroke == False):
                for myTrump in myColorCards[trump]:
                    isBroke = True
                    myColorCards[trump].remove(myTrump)
                    break
            if (isBroke == False):
                return False
    return True

```

```

result = CanWin()

```

```

out = open("output.txt", "w")
if (result):
    out.write("YES")
else:
    out.write("NO")
out.close()

```

тесты пройдены:

Activities Google Chrome May 4 11:04

acmp.ru/index.asp?main=status [English] 5/4/2022, 11:03:46 AM

## Школа программиста

новости | задачи | курсы | олимпиады

информация

- О школе
- Правила
- Олимпиады
- Фотоальбом
- Гостевая
- Форум
- Паспорт
- Мои задачи
- Балас
- Архив олимпиад

задания

- Архив задач
- Состояние системы
- Рейтинг
- Курсы

методика

- Новичкам
- Работа в системе
- Курсы ККДП
- Дистрибутивы
- Статьи
- Ссылки

статистика

балайн

Забери призы в игре

### СОСТОЯНИЕ СИСТЕМЫ

№1 - №20

ID	Дата	Автор	Задача	Язык	Результат	Тест	Время	Память
17088792	04.05.2022 11:03:59	Жазитова Мерей	0892	Java	Accepted		0,04	1216 Кб
17088791	04.05.2022 11:03:57	Амангелдиев Кайсар	0061	C++	Compilation error			
17088790	04.05.2022 11:03:42	Коркина Анна Михайловна	0698	Python	Accepted		0,046	358 Кб
17088789	04.05.2022 11:03:37	Рудович Ренат Евгеньевич	1127	C++	Accepted		0,015	420 Кб
17088788	04.05.2022 11:03:31	Отенберген Айбек	0003	Java	Accepted		0,056	1224 Кб
17088787	04.05.2022 11:03:17	Лизунов Александр Андреевич	0081	C++	Wrong answer	1		408 Кб
17088786	04.05.2022 11:03:13	Праздничных Григорий Олегович	0757	Python	Wrong answer	10	0,046	358 Кб
17088785	04.05.2022 11:02:43	Амантай Мерей	0328	Java	Accepted		0,134	1220 Кб
17088784	04.05.2022 11:02:36	Пятибратов А В	1404	Python	Wrong answer	2	0,015	358 Кб
17088783	04.05.2022 11:02:16	Лизунов Александр Андреевич	0081	C++	Wrong answer	1		408 Кб
17088782	04.05.2022 11:02:16	Хабидуллозада Наджибулло	1418	C++	Wrong answer	13	0,015	420 Кб
17088781	04.05.2022 11:01:54	Алибек	0680	C++	Accepted		0,015	412 Кб
17088780	04.05.2022 11:01:53	Цой Максим Андреевич	0784	C++	Wrong answer	4	0,015	416 Кб
17088779	04.05.2022 11:01:36	Амантай Мерей	0328	Java	Wrong answer	17	0,071	1216 Кб
17088778	04.05.2022 11:01:33	Каширина Олеся Сергеевна	0411	C++	Wrong answer	2	0,015	416 Кб
17088777	04.05.2022 11:01:04	Лизунов Александр Андреевич	0081	C++	Wrong answer	1		408 Кб
17088776	04.05.2022 11:00:59	Сорока Артём Сергеевич	0007	C++	Presentation error	1		408 Кб
17088775	04.05.2022 11:00:58	Новикова Анастасия Юрьевна	0913	Visual C++	Wrong answer	24	0,062	324 Кб
17088774	04.05.2022 11:00:45	Ломать Марина Александровна	1214	Free Pascal	Accepted		0,015	596 Кб
17088773	04.05.2022 11:00:37	Жебрик Евгений Андреевич	0021	C++	Wrong answer	1		400 Кб

№1 - №20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 >>

Вперед >

Яндекс Директ

Получите промокод 5000 рублей на первую рекламную кампанию в Яндекс.Директе

5000 руб.

Получить

Данная задача решается при помощи жадных алгоритмов, изначально мы сортируем наши карты по возрастанию и карты противника по убыванию. Далее мы проходимся по козырям противника и пытаемся их отбить (при поиске карты для биты мы берём минимальную карту, которая превосходит карту противника). Далее мы проходимся по всем не козырным картам противника, и бьём их картами той же масти, если же мы не можем побить мы обращаемся к нашим козырям, если же их нет, то мы не можем побить и возвращается False. Если же все карты побиты, то возвращаем True.

## Вывод:

Для большинства задач наиболее оптимальными решениями будут — жадный алгоритм и динамическое программирование. Так же в некоторых ситуациях можно обойтись и без прямого использования динамики или жадных алгоритмов, но скорость написания и размер такого кода будет уступать предыдущим вариантам.