

Информатика. 27 задание. Теория.

Пояснение: нижеописанное будет сказано для питона, но во многом применимо в других языках своими схожими синтаксическими конструкциями

Первый прототип, который будет разобран имеет примерно следующую формулировку:

База-прототип: Дано целое число n , а затем подается последовательность из n натуральных чисел.

Требуется найти (количество, сумму, разность, макс, мин) среди пар чисел, которые идут либо подряд ($a[i]$, $a[i+1]$), либо идут как угодно.

Задача 1

Дано целое число n , а затем подается последовательность из n натуральных чисел. Требуется найти максимальную сумму, состоящую из пары чисел (парой являются любые два различных элемента), которая будет четна.

Пример входных данных:

5
2
10
5
17
3

Выходные данные для приведённого выше примера: 22

Решение.

```
n = int(input())
summ = 0
max_kr2 = 0
max_nekr2 = 0
for i in range(n):
    x = int(input())
```

```
if x % 2 == 0:
    summ = max(summ, x+max_kr2)
    max_kr2 = max(x, max_kr2)
else:
    summ = max(summ, x+max_nekr2)
    max_nekr2 = max(x, max_nekr2)
print(summ)
```

Задача 2

Дано целое число n , затем последовательность из n натуральных чисел. Требуется найти максимальное произведение, состоящее из пары чисел, которое будет кратно 6.

Пример входных данных:

5
2
10
5
17
3

Выходные данные для приведённого выше примера: 30

Решение.

```
n = int(input())
max_kr_6, kr_3 = 0, 0
kr_2, nekr = 0, 0
ans = 0
for i in range(n):
    x = int(input())
    if x % 6 == 0:
        ans = max(ans, x*max_kr_6, x*kr_3, x*kr_2, x*nekr)
        max_kr_6 = max(x, max_kr_6)
    elif x % 3 == 0:
        ans = max(ans, max_kr_6*x, x*kr_2)
        kr_3 = max(x, kr_3)
    elif x % 2 == 0:
        ans = max(ans, max_kr_6*x, x*kr_3)
        kr_2 = max(x, kr_2)
    else:
        ans = max(ans, x*max_kr_6)
        nekr = max(x, nekr)
print(ans)
```

Задача 3

Дано целое число n , затем последовательность из n натуральных чисел. Требуется найти количество пар чисел, сумма которых будет кратна 91.

Пример входных данных:

5
2
10
5
17
3

Выходные данные для приведённого выше примера: 2

Решение.

```
n = int(input())
ans = 0
a = [0]*91
for i in range(n):
    x = int(input())
    ans += a[(91-x)%91]
    a[x%91] += 1
print(ans)
```

Задача 4

Дано целое число n , затем последовательность из n натуральных чисел. Требуется найти количество пар чисел, сумма которых будет кратна 8, произведение кратно 7.

Пример входных данных:

5
7
25
5
17
3

Выходные данные для приведённого выше примера: 2

Решение.

```
#Динамика
n = int(input())
kr_7 = [0]*8
nekr = [0]*8
ans = 0
for i in range(n):
    x = int(input())
    if x % 7 == 0:
        ans += kr_7[(8-x%8)%8] + nekr[(8-x%8)%8]
        kr_7[x%8] += 1
    else:
        ans += kr_7[(8 - x % 8) % 8]
        nekr[x % 8] += 1
print(ans)
```

Теперь посмотрим на задачи, в которых добавлено одно маленькое, но немаловажное условие — пары находятся на расстоянии не меньше, чем какое-то (то есть разница в индексах по модулю больше или равна чему-то).

База-прототип:

Дано целое число n затем последовательность из n натуральных чисел.

Требуется найти

количество/макс./мин. сумму пар,

чья сумма/разность/произведение кратна(о) чему-то,

а расстояние между элементами пары не меньше k .

Решим следующую задачу неэффективно:

Дано целое число n затем последовательность из n натуральных чисел. Требуется найти количество пар, чья сумма кратна 8, а расстояние между элементами пары не меньше 3.

Пример входных данных:

7

17

15

1

5

123

1

38

Для таких входных данных значением искомого суммы будет число 1

Решение.

```
n = int(input())
ans = 0
a = []
for i in range(n):
    a += [int(input())]
for i in range(n):
    for j in range(i+3, n):
        if (a[i]+a[j]) % 8 == 0:
```

```
ans += 1  
  
print(ans)
```

Итак, рассмотрим эффективное решение предыдущей задачи:

Задача 5

Дано целое число n затем последовательность из n натуральных чисел. Требуется найти количество пар, чья сумма кратна 8, а расстояние между элементами пары не меньше 3.

Решение.

```
n = int(input())
line = []
a = [0]*8
ans = 0
for i in range(3):
    line.append(int(input()))
for i in range(3, n):
    #1й блок: Отправляем на свалку
    #Самый крайний элемент, с которым x может взаимодействовать
    a[line[0] % 8] += 1
    #2й блок: Взаимодействие всех подходящих с x
    x = int(input())
    ans += a[(8 - x%8)%8]
    #3й блок: Сдвиг очереди
    for j in range(3-1):
        line[j] = line[j+1]
    line[2] = x
print(ans)
```

Разберем задачи следующего типа:

База-прототип:

Дано число n затем n строк,
каждой по паре натуральных различных чисел.
Из каждой пары берется одно число так,
чтобы итоговая сумма всех таких чисел
была кратна/некратна *целому числу* и была бы макс/мин.

В целом, задачи по сложности и по "идейности" нужно разделять на два блока:

- 1) Задачи, про сумму некратную чему-либо
- 2) Задачи, про сумму кратную чему-либо

Задачи первого типа:

Задача 6

Дано число n затем n строк, в каждой по паре натуральных различных чисел. Из каждой пары берется одно число так, чтобы итоговая сумма всех таких чисел была некратна 10 и была бы:

- а) Максимальна
- б) Минимальна

Пример входных данных:

```
3
10 20
21 11
1 1
```

Для таких входных данных значением искомой суммы будет число 42 и 22

Решение. Пункт а)

```
n = int(input())
ans = 0
diff = 10000000000
for i in range(n):
    a, b = map(int, input().split())
    #Ниже альтернативный способ считать два числа:
    #a = [*что делаем,* откуда делаем,* при каком условии*]
    #a = [int(i) for i in input().split()]
```

```
ans += max(a,b)
if abs(a-b) % 10 != 0:
    diff = min(diff, abs(a-b))
print(ans - diff * (ans % 10 == 0))
```

Решение пункта б) выглядит следующим образом:

Решение. 6)

```
#Мин. сумма
n = int(input())
ans = 0
diff = 10000000000
for i in range(n):
    a, b = map(int, input().split())
    ans += min(a,b)
    if abs(a-b) % 10 != 0:
        diff = min(diff, abs(a-b))
print(ans + diff * (ans % 10 == 0))
```

Разберем задачи второго типа:

Задача 7

Дано число n затем n строк, в каждой по паре натуральных различных чисел. Из каждой пары берется одно число так, чтобы итоговая сумма всех таких чисел была кратна 2 и максимальна.

Пример входных данных:

```
3
10 20
21 11
1 1
```

Для таких входных данных значением искомого суммы будет число 42

Решение.

```
n = int(input())
ans = 0
diff = 10000000000
for i in range(n):
    a = [int(i) for i in input().split()]
    ans += max(a)
    if abs(a[0]-a[1]) % 2 == 1:
        diff = min(diff, abs(a[0]-a[1]))
print(ans - diff * (ans % 2 == 1))
```

Задача 8

Дано число n затем n строк, в каждой по паре натуральных различных чисел. Из каждой пары берется одно число так, чтобы итоговая сумма всех таких чисел была кратна 3 и минимальна.

Пример входных данных:

```
3
13 20
21 11
1 1
```

Для таких входных данных значением искомого суммы будет число 25

Решение.

```
n = int(input())
ans = 0
#инд 0 и 1 отвечают за мин дифы с ост. 1
#2 и 3 за мин дифы с ост. 2
diff = [1000000]*4
for i in range(n):
    a, b = map(int, input().split())
    ans += min(a,b)
    tmp = abs(a-b)
    if tmp % 3 == 1:
        if tmp < diff[0]:
            diff[1], diff[0] = diff[0], tmp
        else:
            diff[1] = min(diff[1], tmp)
    if tmp % 3 == 2:
        if tmp < diff[2]:
            diff[3], diff[2] = diff[2], tmp
        else:
            diff[3] = min(diff[3], tmp)
if ans % 3 == 0:
    print(ans)
elif ans % 3 == 1:
```

```
    print(ans + min(diff[2], diff[0]+diff[1]))  
else:  
    print(ans + min(diff[0], diff[2] + diff[3]))
```

Задача 9

Дано число n затем n строк, в каждой по паре натуральных различных чисел. Из каждой пары берется одно число так, чтобы итоговая сумма всех таких чисел была кратна 10 и максимальна.

Пример входных данных:

```
3
10 20
21 11
1 9
```

Для таких входных данных значением искомого суммы будет число 50

Решение.

```
n = int(input())
ans = [0]*10 #Массив всех макс. сумм, которые имеют опр. ост. при дел. на 10
for i in range(n):
    a, b = map(int, input().split())
    #Создаем копию массива, чтобы не повредить данные изначального
    #a также адекватно посчитать новые подх. суммы.
    ans_new = [-1000000000]*10

    for j in range(10): #Блок подсчета всех сумм, путем прибавления числа a
        ost = (ans[j] + a) % 10
        #Сравниваем с ans_new, потому что изначально он забит мусором
        #И если мы нашли сумму, то она уже будет больше.
        #Если сравнивать с ans, то есть риск получить промежуточную сумму
        #неадекватного значения
        if ans[j] + a > ans_new[ost]:
            ans_new[ost] = ans[j] + a

    for j in range(10): #Блок подсчета всех сумм, путем прибавления числа b
        ost = (ans[j] + b) % 10
        if ans[j] + b > ans_new[ost]:
            ans_new[ost] = ans[j] + b
```

```
ans = ans_new #Обновление всех макс. сумм, имеющих опр. ост. при дел. на 1
print(ans[0])
```

Для закрепления разберем еще одну задачу, относительно упростив код:

Задача 10

Дано число n затем n строк, в каждой по тройке натуральных различных чисел. Из каждой тройки берется одно число так, чтобы итоговая сумма всех таких чисел была кратна 8 и была бы минимальна.

Пример входных данных:

3

8 20

21 11

5 13

Для таких входных данных значением искомого суммы будет число 24

Решение.

```
ans = [0]*8
n = int(input())
for i in range(n):
    a = [int(j) for j in input().split()]
    ans_new = [100000000]*8
    for k in range(3):
        for j in range(8):
            ost = (a[k] + ans[j]) % 8
            ans_new[ost] = min(ans_new[ost], a[k] + ans[j])
    ans = ans_new
print(ans[0])
```

Рассмотрим также следующий тип задач:

Задача 11

Дано число n затем n строк, в каждой по тройке натуральных различных чисел. Найдите три суммы таких, чтобы минимальная была нечетна, а максимальная четна. Каждое число обязательно прибавляется в одну из сумм и только одно. Выведите макс. сумму, удовлетворяющую вышеописанным условиям. Подразумевается, что подобную сумму получить можно.

Решение.

```
n = int(input())
s = [0]*3 #0 - мин, 1 - ср., 2 - макс.
#10 20 30
#diff_0 = средний - мин.
#diff_1 = средний - макс.
#diff_2 = макс. - мин.
diff = [10000000]*3

for i in range(n):
    a = sorted(list(map(int, input().split())))
    for j in range(3):
        s[j] += a[j]
    if abs(a[0]-a[1]) % 2 == 1:
        diff[0] = min(diff[0], abs(a[0]-a[1]))
    if abs(a[2]-a[1]) % 2 == 1:
        diff[1] = min(diff[0], abs(a[2]-a[1]))
    if abs(a[0]-a[2]) % 2 == 1:
        diff[2] = min(diff[2], abs(a[0]-a[2]))

if s[0] % 2 != 0 and s[2] % 2 == 0:
    print(s[2])

elif s[0] % 2 == 0 and s[2] % 2 == 0:
```

```
if diff[0] != 10000000:
    print(s[2])
else:
    print(s[2]-diff[2]-diff[1])

elif s[0] % 2 != 0 and s[2] % 2 != 0:
    if diff[1] != 10000000 and diff[0] != 10000000 and diff[2] != 10000000:
        print(s[2] - min(diff[1], diff[2]))
    elif diff[1] != 10000000:
        print(s[2] - diff[1])
    elif diff[0] != 10000000 and diff[2] != 10000000:
        print(s[2] - diff[2])

else:
    if diff[1] != 10000000 and diff[0] != 10000000 and diff[2] != 10000000:
        print(s[2] - min(diff[1], diff[2]))
    elif diff[1] != 10000000:
        print(s[2] - diff[1])
    elif diff[0] != 10000000 and diff[2] != 10000000:
        print(s[2] - diff[2])
```

Также разберем задачи на префиксные суммы:

База-прототип: Дана последовательность натуральных чисел. Необходимо найти максимально возможную сумму её непрерывной подпоследовательности, кратную чему-либо.

На конкретной задаче:

Задача 12

Дана последовательность натуральных чисел. Необходимо найти максимально возможную сумму её непрерывной подпоследовательности, кратную 10. Выведите наибольшую длину последовательности, при которой сумма будет кратна 10 и максимальна.

Первая строка содержит натуральное число N — общее количество чисел в наборе. Каждая из следующих N строк содержит одно число.

Пример:

7
23
13
4
11
6
19
8

Ответом для примера будет: 4

Решение.

```
n = int(input())
s = [1000000]*10
s[0] = 0
summ = 0
ans = 0
ind = [-1]*10
dlina = 0

for i in range(n):
    summ += int(input())

    if summ - s[summ % 10] > ans:
        ans = summ - s[summ % 10]
        dlina = i - ind[summ % 10]

    elif summ - s[summ % 10] == ans:
        dlina = max(dlina, i - ind[summ % 10])

    if summ < s[summ % 10]:
        s[summ % 10] = summ
        ind[summ % 10] = i

print(dlina)
```

Рассмотрим задачу, в которой нужно придумать, как именно использовать префиксные суммы:

Задача 13

Дана последовательность натуральных чисел. Рассматриваются все её непрерывные подпоследовательности, в которых количество простых чисел кратно $K = 3$. Найдите наибольшую сумму такой подпоследовательности.

Пример:

7

23

13

4

11

6

19

8

Ответом для примера будет: 61

Рассмотрим пример из условия.

В этом наборе можно выбрать две непрерывные последовательности, содержащие по 3 простых числа ($23+13+4+11+6=57$) и ($13+4+11+6+19+8=61$).

Решение.

```
def prime(x):
    for i in range(2, int(x**0.5)+1):
        if x % i == 0:
            return False
    return True

n = int(input())
data = [] #Все числа из файла
num = [0] * n #Индекс этого массива равен индексу data и показывает, явл. ли эл-ты из data
               #Простыми числами. Если да - в массиве будет стоять 1.
for i in range(n):
    data += [int(input())]
    if prime(data[i]):
        num[i] += 1

summa = 0 #Сумма всех единиц массива num
s = [1000000]*3 #Мин. частичные суммы, имеющие опр. остаток при делении на 3.
               #Почему на 3? Потому кол-во простых чисел должно быть кр. 3.
s[0] = 0
ind = [-1]*3 #На каком индексе была получена мин. частичная сумма
ind[0] = 0
dlna = 0 #Длина последовательности, в которой сумма наибольшая
kr = 0 #Число, отражающее макс. кол-во простых чисел в рассматриваемой подпослед.,
       #которое будет кр. 3
ans = 0

for i in range(n):
    summa += num[i] #Подсчет i-ой частичной суммы из массива 1 и 0
    if summa - s[summa % 3] > 0:
        ans = max(ans, sum( data[ind[summa % 3]:i+1] ))
    if summa < s[summa % 3]: #Обновление мин. частичных сумм
        s[summa % 3] = summa #обновляем частичную сумму
        ind[summa % 3] = i    #Индекс последнего элемента этой суммы

print(ans)
```

Также рассмотрим задачу с досрока ЕГЭ 2022 года и ее решение:

Задача 14

В городе М расположена кольцевая автодорога длиной в N километров с движением в обе стороны. На каждом километре автодороги расположены пункты приема мусора определенной вместимости. В пределах кольцевой дороги в одном из пунктов сборки мусора собираются поставить мусороперерабатывающий завод таким образом, чтобы стоимость доставки мусора была минимальной. Стоимость доставки мусора вычисляется, как вместимость пункта сбора умноженная на расстояние от пункта сбора мусора до мусороперерабатывающего завода. Если мусороперерабатывающий завод находится рядом с пунктом сбора расстояние считается нулевым. Контейнеры нумеруются с 1 до N. Рядом с каким пунктом сбора мусора нужно поставить мусороперерабатывающий завод?

Описание входных данных:

Первое число N — количество контейнеров для мусора. Последующие N чисел — количество килограмм мусора, которое производится на точке.

Описание выходных данных:

Одно число — номер контейнера для мусора рядом с которым стоит расположить перерабатывающий завод.

Решение.

```
file = open('test.txt', 'rt')
n = int(file.readline())
a = [int(file.readline()) for i in range(n)]
s = [sum(a[0:n // 2])]
for i in range(1, n):
    s.append(s[i - 1] - a[i - 1] + a[(i - 1 + n // 2) % n])

summa = sum(a)
p = []
for i in range(n):
    p.append(summa - s[i])

price = 0
```

```
for i in range(n // 2):  
    price += i * a[i]  
for i in range(n // 2, n):  
    price += (n - i) * a[i]
```

```
minim = price  
min_k = -1  
for i in range(1, n):  
    price -= s[i]  
    price += p[i]  
    if price < minim:  
        minim = price  
        min_k = i + 1  
print(min_k)
```