

Линейные алгоритмы

Горденко Мария Константиновна

Линейные алгоритмы

Линейные алгоритмы – класс алгоритмов, позволяющие решать задачи за линейную сложность $O(n)$, где n – размерность задачи

Задача №6: Поиск пропущенного значения

Дан массив a из $n - 1$ элементов, $\forall i = \overline{0, n - 2}: a[i] \in \{1, \dots, n\}$,
 $\forall i, j = \overline{0, n - 2}: i \neq j \Leftrightarrow a[i] \neq a[j]$. Требуется найти число
 $1 \leq k \leq n$, не встречающееся в a

Задача №6: Поиск пропущенного значения

Дан массив a из $n - 1$ элементов, $\forall i = \overline{0, n - 2}: a[i] \in \{1, \dots, n\}$,
 $\forall i, j = \overline{0, n - 2}: i \neq j \Leftrightarrow a[i] \neq a[j]$. Требуется найти число
 $1 \leq k \leq n$, не встречающееся в a

Решение:

1. Вычислим сумму чисел от 1 до n
как сумму арифметической
прогрессии: $S_n = \frac{a_1 + a_n}{2} n = \frac{n(n+1)}{2}$
2. Вычтем из суммы элементы a
3. Разность – пропущенное число

Задача №6: Поиск пропущенного значения

Дан массив a из $n - 1$ элементов, $\forall i = \overline{0, n - 2}: a[i] \in \{1, \dots, n\}$,
 $\forall i, j = \overline{0, n - 2}: i \neq j \Leftrightarrow a[i] \neq a[j]$. Требуется найти число
 $1 \leq k \leq n$, не встречающееся в a

Решение:

1. Вычислим сумму чисел от 1 до n как сумму арифметической прогрессии: $S_n = \frac{a_1 + a_n}{2} n = \frac{n(n+1)}{2}$
2. Вычтем из суммы элементы a
3. Разность – пропущенное число

```
def findMissingElement(a):  
    n = len(a) + 1  
    sum_n = n * (n + 1) // 2  
    for value in a:  
        sum_n -= value  
    return sum_n
```

Задача 6.1

- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.
- Ввод: 1 2 2 1 2 1 3
- Вывод: 3, 2
- Ввод: 1 2 10 10 10 1 1
- Вывод: 2, 1

Задача 6.1. Обсуждение

- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.
- Ввод: 1 2 2 1 2 1 3 3
- Вывод: 3, 2
- Ввод: 1 2 10 10 10 1 1
- Вывод: 2, 1

Задача 6.1. Решение

- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.

- Ввод: 1 2 2 1 2 1 3

- Вывод: 3, 2

- Ввод: 1 2 10 10 10 1 1

- Вывод: 2, 1

```
l = [1, 2, 2, 1, 2, 1, 3, 3]
bin = [0] * 32
for i in range(len(l)):
    num = l[i]
    j = 0
    while (num != 0):
        bin[j] += num % 2
        num //= 2
        j += 1
```

```
num = 0
pow_2 = 1
count = 0
for i in range(len(bin)):
    bin[i] %= 3
    if (bin[i] > 0):
        num += pow_2
        count = bin[i]
    pow_2 *= 2
print(num, count)
```


Задача №7: Поиск уникального элемента

Дан массив чисел a , все из которых, кроме b , встречаются чётное число раз. Нужно найти число b , встречающееся нечётное число раз

Задача №7: Поиск уникального элемента

Дан массив чисел a , все из которых, кроме b , встречаются чётное число раз. Нужно найти число b , встречающееся нечётное число раз

Решение:

Воспользуемся операцией *xor* (\oplus). Вспомним её свойства:

- $k \oplus 0 = k$
- $k \oplus k = 0$
- $(k \oplus l) \oplus l = k$

Представим *xor* значений a так:

$$(a[0] \oplus a[0]) \oplus \dots \oplus (a[n] \oplus a[n]) \oplus b = 0 \oplus 0 \oplus \dots \oplus 0 \oplus b = 0 \oplus b$$

Задача №7: Поиск уникального элемента

Дан массив чисел a , все из которых, кроме b , встречаются чётное число раз. Нужно найти число b , встречающееся нечётное число раз

Решение:

Воспользуемся операцией *xor* (\oplus). Вспомним её свойства:

- $k \oplus 0 = k$
- $k \oplus k = 0$
- $(k \oplus l) \oplus l = k$

Представим *xor* значений a так:

$$(a[0] \oplus a[0]) \oplus \dots \oplus (a[n] \oplus a[n]) \oplus b = 0 \oplus 0 \oplus \dots \oplus 0 \oplus b = 0 \oplus b$$

```
def findUniqueElement(a):  
    b = 0  
    for value in a:  
        b ^= value  
    return b
```

Задача №8: Поиск элемента массива, встречающегося чаще всех остальных

Дан массив a из n элементов. Найти тот, что повторяется $> \frac{n}{2}$ раз

Задача №8: Поиск элемента массива, встречающегося чаще всех остальных

Дан массив a из n элементов. Найти тот, что повторяется $> \frac{n}{2}$ раз

Решение:

Переменные $value$ и $count$ хранят элемент массива и число его повторений. Для очередного элемента возможны три варианта:

- $count = 0 \Rightarrow value = a[i]$
 $count += 1$
- $a[i] = value \Rightarrow count += 1$
- $a[i] \neq value \Rightarrow count -= 1$

После просмотра всех элементов, в $value$ окажется искомое число

Задача №8: Поиск элемента массива, встречающегося чаще всех остальных

Дан массив a из n элементов. Найти тот, что повторяется $> \frac{n}{2}$ раз

Решение:

Переменные $value$ и $count$ хранят элемент массива и число его повторений. Для очередного элемента возможны три варианта:

- $count = 0 \Rightarrow value = a[i]$
 $count += 1$
- $a[i] = value \Rightarrow count += 1$
- $a[i] \neq value \Rightarrow count -= 1$

```
def findMostFrequent(a):  
    answer = 0  
    count = 0  
    for value in a:  
        if count == 0:  
            answer = value  
            count += 1  
        elif answer == value:  
            count += 1  
        else:  
            count -= 1  
    return answer
```

После просмотра всех элементов, в $value$ окажется искомое число

Задача №9: Подмассив с заданной суммой

Дан массив a из n элементов ($\forall i = \overline{0, n-1}: a[i] \in \mathbb{N}$). Требуется найти подмассив, сумма элементов которого равна k

Задача №9: Подмассив с заданной суммой

Дан массив a из n элементов ($\forall i = \overline{0, n-1}: a[i] \in \mathbb{N}$). Требуется найти подмассив, сумма элементов которого равна k

Решение:

Пусть l и r – границы искомого подмассива

Для каждого r пытаемся набрать сумму k .

Двигаем l , пока текущая сумма $> k$.

Если для текущего r набрать не удалось,
то будем рассматривать следующее r

Метод решения задачи – метод **двух указателей** – границ зоны

Задача №9: Подмассив с заданной суммой

Дан массив a из n элементов ($\forall i = \overline{0, n-1}: a[i] \in \mathbb{N}$). Требуется найти подмассив, сумма элементов которого равна k

Решение:

Пусть l и r – границы искомого подмассива

Для каждого r пытаемся набрать сумму k .

Двигаем l , пока текущая сумма $> k$.

Если для текущего r набрать не удалось, то будем рассматривать следующее r

Метод решения задачи – метод **двух указателей** – границ зоны

```
def findSubsequence(a, k):  
    l = 0  
    sum_ = 0  
    for r in range(len(a)):  
        sum_ += a[r]  
        while sum_ > k:  
            sum_ -= a[l]  
            l += 1  
        if sum_ == k:  
            break  
    return [l, r]
```

Скольльзящее окно

- **Задача.** Для массива, состоящего из N целых чисел, найдите непрерывный подмассив заданной длины k , который имеет максимальное среднее значение. Нужно вывести максимальное среднее значение.

.

Скользящее окно

- $n = 6, k = 3, nums = [4, 2, 6, 7, 6, 8]$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 4 + 2 + 6 = 12,$
 $result = sum = 12$

Скользящее окно

- $n = 6, k = 3, nums = [4, 2, 6, 7, 6, 8]$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 4 + 2 + 6 = 12,$
 $result = sum = 12$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 12 - 4 + 7 = 15,$
 $result < sum, result = 15$

Скользящее окно

- $n = 6, k = 3, nums = [4, 2, 6, 7, 6, 8]$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 4 + 2 + 6 = 12,$
 $result = sum = 12$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 12 - 4 + 7 = 15,$
 $result < sum, result = 15$

0	1	2	3	4	5
4	2	6	7	6	8

$sum = 15 - 2 + 6 = 19,$
 $result < sum, result = 19$

Скользящее окно

- $n = 6, k = 3, nums = [4, 2, 6, 7, 6, 8]$

0	1	2	3	4	5
4	2	6	7	6	8

$$sum = 4 + 2 + 6 = 12,$$
$$result = sum = 12$$

0	1	2	3	4	5
4	2	6	7	6	8

$$sum = 12 - 4 + 7 = 15,$$
$$result < sum, result = 15$$

0	1	2	3	4	5
4	2	6	7	6	8

$$sum = 15 - 2 + 6 = 19,$$
$$result < sum, result = 19$$

0	1	2	3	4	5
4	2	6	7	6	8

$$sum = 15 - 6 + 8 = 17,$$
$$result > sum, result = 19$$

$$\text{Ответ: } \frac{result}{k} = \frac{19}{3}$$