

**Министерство науки и высшего образования Российской
Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет цифровых трансформаций

Дисциплина:
«Алгоритмы и структуры данных»

Практическая работа №7
«Жадные алгоритмы»

Выполнила:
Абаянцева Е. Ю., студент группы J3110

Санкт-Петербург
2024 г.

1. Подсчёт памяти: два массива 16 байт, один `int` – 4 байта. Вызов `k` раз (на запускаемом тесте – 5 раз) с подсчётом памяти каждой из трёх функций (в сумме три массива в каждой функции – 24 байта, два `int` в двух – 8 байтов). Всего в сумме – 361 байт.
2. Подсчёт асимптотики – сложность $O(k \cdot (m + n))$.
3. Код:

```

7  class Solution {
8  public:
9      vector<int> maxnumber(vector<int>& nums1, vector<int>& nums2, int k) {
10         int m = nums1.size(), n = nums2.size();
11         vector<int> nums;
12         for (int i = max(0, k - n); i <= min(k, m); i++) {
13             nums = max(nums, mergenums(maxnums(nums1, i), maxnums(nums2, k - i)));
14         }
15         return nums;
16     }
17
18     vector<int> maxnums(vector<int>& nums, int k) {
19         int i = (int)nums.size() - k;
20         vector<int> end_nums;
21         for (int num : nums) {
22             while (i > 0 && end_nums.size() > 0 && end_nums.back() < num) {
23                 end_nums.pop_back();
24                 i--;
25             }
26             end_nums.push_back(num);
27         }
28         end_nums.resize(k);
29         return end_nums;
30     }
31
32     vector<int> mergenums(vector<int> nums1, vector<int> nums2) {
33         vector<int> nums;
34         while (nums1.size() > 0 || nums2.size() > 0) {
35             if (nums1 > nums2) {
36                 vector<int> &mas = nums1;
37                 nums.push_back(mas[0]);
38                 mas.erase(mas.begin());
39             }
40             else {
41                 vector<int> &mas = nums2;
42                 nums.push_back(mas[0]);
43                 mas.erase(mas.begin());
44             }
45         }
46         return nums;
47     }
48 };
49
50
51 int main() {
52     vector<int> nums1 = { 3, 4, 6, 5 };
53     vector<int> nums2 = { 9, 1, 2, 5, 8, 3 };
54     int k = 5;
55     Solution x;
56     vector<int> result = x.maxnumber(nums1, nums2, k);
57     for (int num : result) {
58         cout << num << " ";
59     }
60     return 0;
61 }

```

Рис. 1: КОД

4. Прохождение всех локальных тестов:
 - (а) Тест 1:

Accepted Runtime: 0 ms

• **Case 1** • Case 2 • Case 3

Input

```
nums1 =  
[3,4,6,5]
```

```
nums2 =  
[9,1,2,5,8,3]
```

```
k =  
5
```

Output

```
[9,8,6,5,3]
```

Expected

```
[9,8,6,5,3]
```

Рис. 2: ТЕСТ 1

(b) Тест 2:

Accepted Runtime: 0 ms

• Case 1 • **Case 2** • Case 3

Input

nums1 =
[6,7]

nums2 =
[6,0,4]

k =
5

Output

[6,7,6,0,4]

Expected

[6,7,6,0,4]

Рис. 3: ТЕСТ 2

(с) Тест 3:

Accepted Runtime: 0 ms

• Case 1 • Case 2 • **Case 3**

Input

```
nums1 =  
[3,9]
```

```
nums2 =  
[8,9]
```

```
k =  
3
```

Output

```
[9,8,9]
```

Expected

```
[9,8,9]
```

Рис. 4: ТЕСТ 3

5. Прохождение основных тестов:

Description	Accepted	Editorial	Solutions	Submissions		
Status	Language	Runtime	Memory	Notes		
Accepted 11 hours ago	C++	39 ms	32.4 MB			

Рис. 5: ОСНОВНЫЕ ТЕСТЫ

6. В этой задаче требуется использовать жадный алгоритм, потому что задача требует на каждом шагу выбирать оптимально рабочий вариант (в нашем варианте составить список максимальных элементов).

На каждом шаге мы выбираем массив с максимальными числами, сравнивая его с составленным массивом с таким же количеством элементов, но с другими максимальными элементами (если они существуют).