

Задача 9

Всесторонний анализ алгоритма



```
1 #include <iostream>
2 #include <vector>
3 #include <limits.h>
4
5 int long_find_max_subarray(std::vector<int>& arr,
6     int k) {
7     int n = arr.size();
8     int max_sum = INT_MIN;
9
10    for (int i = 0; i <= n - k; ++i) {
11        int current_sum = 0;
12        for (int j = i; j < i + k; ++j) {
13            current_sum += arr[j];
14        }
15
16        max_sum = std::max(max_sum, current_sum);
17    }
18
19    return max_sum;
20}
```

1. Оцените и обоснуйте асимптотическую верхнюю границу сложности для этой функции.
2. Как можно **оптимизировать** этот алгоритм? Разработайте оптимизированный алгоритм и обоснуйте его сложность.

В худшем случае - $n - k + 1$ операций

В лучшем случае - K

$$T(n, k) = (n - k + 1) k = nk - k^2 + k = \frac{k^{n+1}}{K^n} = O(nk)$$

В среднем случае $K \sim n$

$$T(n, n) = O(n^2)$$