

Алгоритмы и структуры данных | АЗ семинар 3.

```
1 #include <iostream>
2 #include <vector>
3 #include <climits>
4
5 int long_find_max_sum(const 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        max_sum = std::max(max_sum, current_sum);
16    }
17
18    return max_sum;
19 }
20
21 }
```

Асимптотическая сложность
алгоритма - $O(n^2)$.

затратность:

$$\begin{array}{l|l} 1 & \\ 1 & \\ n-k & \\ 1 & \\ i+k & \end{array} \left| \begin{array}{l} i+k \\ 2 \end{array} \right| \sum_{i=0}^{n-k} (i+k+2) = \frac{0+k+2+n+2}{2} (n-k) =$$
$$= \frac{(n+k+4)(n-k)}{2} = \frac{1}{2}(n^2+4n-k^2)$$

$$T(n) = 1 + 1 + \sum_{i=0}^{n-k} (i+k+2) + 2 = \frac{1}{2}(n^2+4n-k^2) + 4 \Rightarrow O(n^2)$$

Оптимизированный алгоритм с сложностью $O(n)$:

```
1 #include <iostream>
2 #include <vector>
3
4 int find_max_sum(const std::vector<int>& arr, int k) {
5
6     int n = arr.size();
7     int curr = 0;
8
9     for (int i = 0; i < k; ++i) {
10         curr += arr[i];
11     }
12
13     int max_sum = curr;
14
15     for (int i = k; i < n - k; ++i) {
16         curr += arr[i] - arr[i - k];
17         max_sum = std::max(a: max_sum, b: curr);
18     }
19
20     return max_sum;
21 }
```

$$\begin{array}{l|l} 1 & \\ 1 & \\ k & \\ i & k \end{array} \left| \begin{array}{l} 1 \\ 3 \\ 2 \end{array} \right| 6(n-k)$$

$$T(n) = 1 + 1 + k + 1 + 6(n-k) = 6n - 5k + 3 \Rightarrow O(n)$$