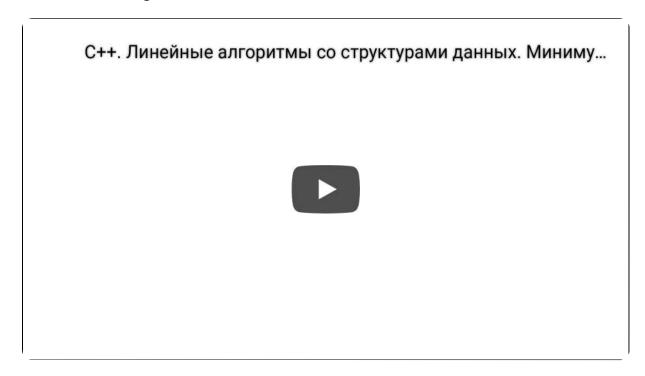
Минимум в скользящем окне



Задача. Для заданного массива чисел $a_1, a_2 \dots a_n$ и числа k необходимо найти минимумы для всех отрезков длины k.

Будем искать минимумы в отрезках по порядку: для первых k элементов, для элементов со 2 до k+1 и так далее. В этой задаче каждые k подряд идущих элементов принято называть окном в исходном массиве. Часто задача встречается под названием "поиск минимума в скользящем окне длиной k".

Заметим, что эту задачу нельзя решить с помощью структуры данных, аналогичной префиксным суммам, или пересчитывая значение минимума в новом окне через значение в предыдущем. Так, например, если в предыдущем окне минимум находился в первом элементе отрезка, то у нас нет никакой информации, помогающей посчитать минимум в текущем окне. В худшем случае такая ситуация будет возникать на каждом шаге (например, для возрастающего массива). Сложность такого алгоритма составит O(k(n-k)).

Рассмотрим решение задачи за линейное время. Запишем в массив ans_i индекс ближайшего элемента справа, меньшего a_i . Мы ранее научились получать значения элементов этого массива за линейное время с помощью стека.

Будем перебирать начала всех окон в цикле по переменной i. Также заведём указатель на текущего кандидата — на минимум в окне imin. Изначально imin равен индексу самого элемента — 1. При рассмотрении окна мы будем менять значение imin на значение ans_{imin} до тех пор, пока ans_{imin} находится внутри окна. Найденный таким образом imin является индексом минимального элемента в рассматриваемом окне. При переходе с предыдущего окна на текущее мы можем оставить imin без изменения, если imin принадлежит и новому окну тоже. В противном случае imin равен самому левому индексу рассматриваемого окна.

Данный алгоритм будет работать за линейное время. Заметим, что imin всегда будет двигаться только вправо, значит, общее количество действий, выполняемых при переборе всех окон, не будет превышать n.

Реализация

```
int imin = 1;
for (int i = 1; i <= n - k + 1; ++i) {
   if (imin < i)
      imin = i;
   while (ans[imin] < i + k)
      imin = ans[imin];
   res[i] = imin;
}</pre>
```