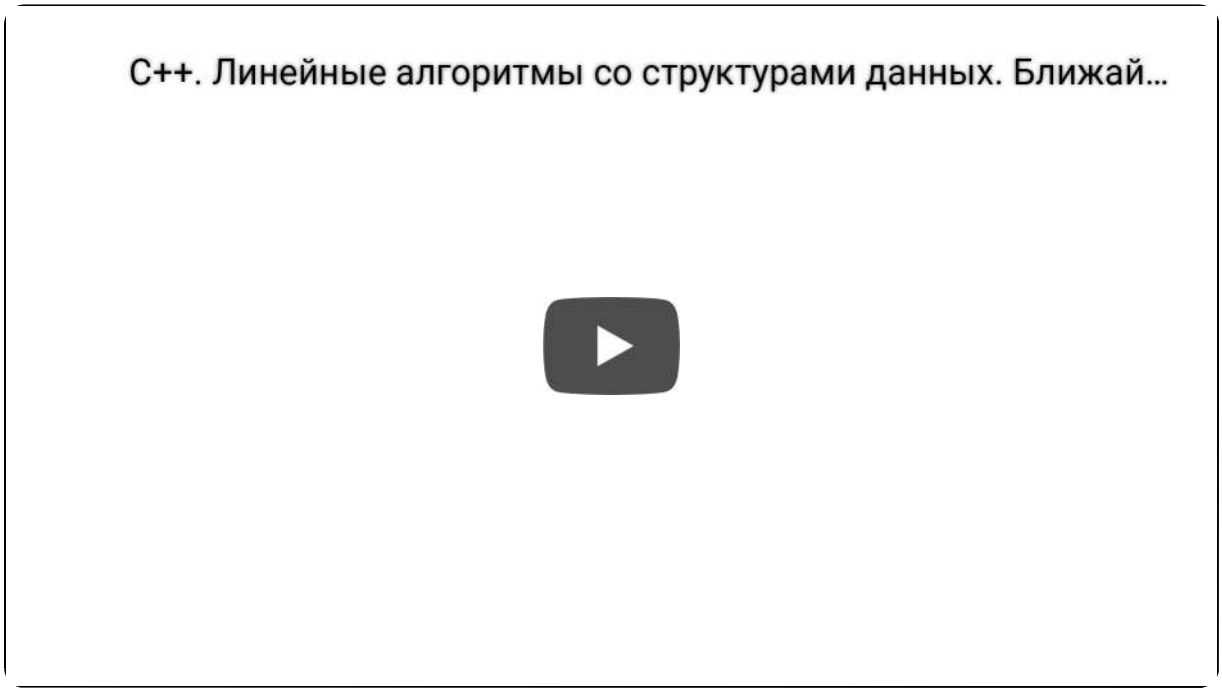
Ближайший из меньших элементов справа



Задача. Дан массив из n элементов — $a_1, a_2, \ldots a_n$. Для каждого элемента массива необходимо найти индекс ближайшего элемента справа, которое меньше него. Если такого элемента не существует, то необходимо вернуть $n + 1$.

Например, для массива $\{3, 6, 8, 4, 2, 5\}$ ответом будет массив: $\{5, 4, 4, 5, 7, 7\}$.

Заметим, что тривиальное решение задачи (поиск ответа по отдельности для каждого элемента) будет иметь вычислительную сложность $O(n^2)$.

Рассмотрим решение данной задачи за линейное время. Для этого будем использовать стек. Будем перебирать все индексы массива слева направо в цикле по переменной i и помещать индексы в стек st, пока значения элементов с этими индексами возрастают. Если текущий элемент с индексом i меньше вершины стека, то будем удалять вершину из стека до тех пор, пока текущий элемент всё ещё меньше вершины. При этом для каждой такой вершины индекс i будет являться ответом. Для удобства реализации алгоритма создадим фиктивный (барьерный) элемент массива a_0 и запишем в

него значение $-\infty$ (число заведомо, меньшее всех значений элементов в массиве). Введение такого элемента гарантирует нам, что стек st никогда не будет пуст. Создадим ещё один фиктивный элемент массива a_{n+1} и сохраним в него также значение $-\infty$. Введение этого элемента позволит корректно обработать элементы, для которых ответа не существует, вернее, в указанной постановке ответом является $n + 1$.

Такой алгоритм будет работать за линейное время, так как каждый элемент добавляется и удаляется из стека не более одного раза.

Реализация

```
int INF = 2e9 + 1;
int n; cin >> n;
vector<int> a(n + 2, -INF), ans(n + 2);
for (int i = 1; i <= n; ++i)
    cin >> a[i]
vector<int> st;
st.push_back(0);
for (int i = 1; i < n + 2; ++i)
{
    while (a[st.back()] > a[i])
    {
        ans[st.back()] = i;
        st.pop_back();
    }
    st.push_back(i);
}
for (int i = 1; i <= n; ++i)
    cout << ans[i] << " ";
cout << endl;
```