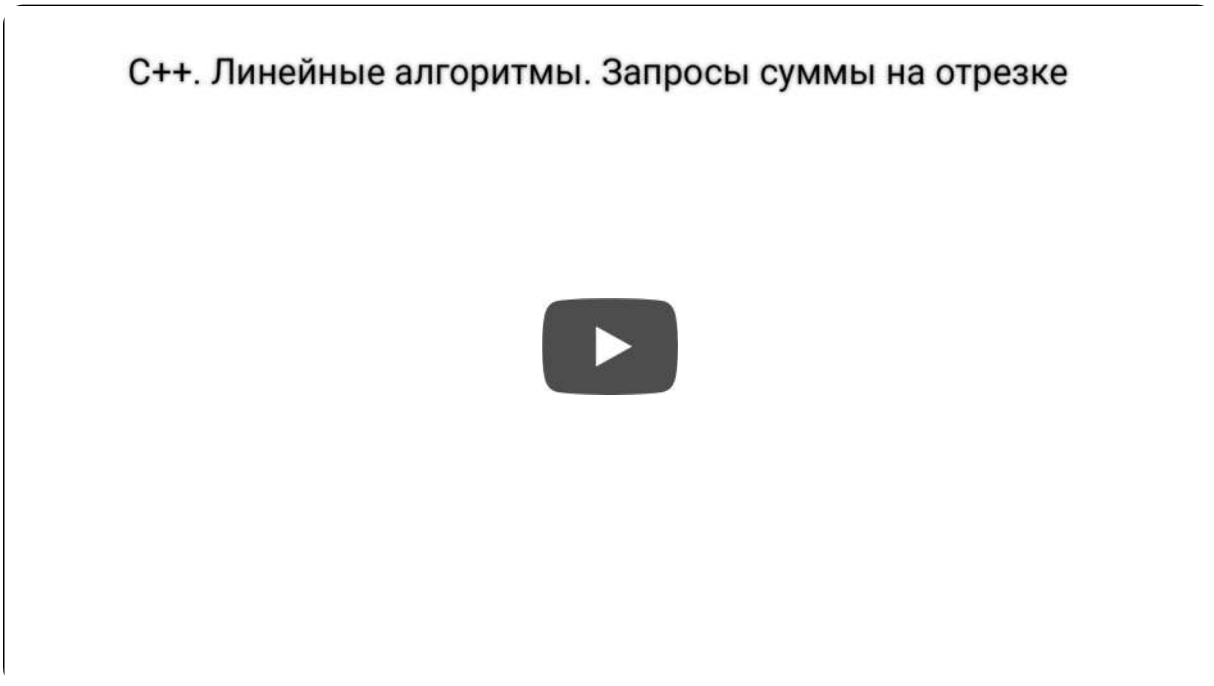
Запросы суммы на отрезке



Задача Пусть дан массив $a_0, a_1, \ldots a_{n-1}$. В элементах массива записано количество осадков, выпавших в моменты времени $0, 1.., n-1$. Необходимо ответить на m запросов о количестве осадков, выпавших с l-го по r-й момент времени $(1 \leq l \leq r \leq n)$.

Формулируя задачу более строго, необходимо уметь отвечать на запросы суммы элементов массива на отрезке с $[l; r]$.

Заметим, что если мы будем на каждый запрос насчитывать циклом сумму элементов, то такое решение будет работать за вычислительную сложность $O(mn)$. Такой алгоритм не будет линейным от входных данных.

Для решения задачи за линейное время рассмотрим структуру данных — массив префиксных сумм. Заведём массив p размера $n+1$. В i-м элементе массива p будем хранить сумму первых i элементов массива a. То есть

- $p_0 = 0$ (сумма 0 элементов массива p)
- $p_1 = a_0$
- $p_2 = a_0 + a_1$
- $p_3 = a_0 + a_1 + a_2$

....

- $p_n = a_0 + a_1 + a_2 + \ldots + a_{n-1}$

Для того чтобы высчитывать массив p эффективно (за линейную сложность), можно воспользоваться следующим соотношением: $p_i = p_{i-1} + a_{i-1}$.

Заметим, что теперь мы можем высчитывать сумму элементов $a_l + a_{l+1} + \ldots + a_r$ как разность $p_{r+1} - p_l$.

Таким образом, мы научились отвечать на запрос суммы элементов на отрезке за $O(1)$ (с предварительным предподсчётом префиксных сумм) и решать всю задачу за $O(n + m)$.

Реализация

```
vector<int>  p(n + 1);
p[0] = 0;
for (int i = 1; i <= n; ++i)
{
    p[i] = p[i - 1] + a[i-1];
}
int m;
cin >> m;
for (int i = 0; i < m; ++i)
{
    int l, r;
    cin >> l >> r;
    cout << p[r+1] - p[l] << endl;
}
```