

# 前缀和引入

杨子凡

Nov 28, 2024

## 1 引入

荀子曰：不积跬步，无以至千里；不积小流，无以成江海。

这句话揭示了世间万物整体和部分之间的关系——庞大的整体由若干个体组成；单个个体虽小，但经过一点一滴的累积，也能聚成一个庞大的整体。学习工作贵在不断积累，不断充实、丰富、完善自己，所有的努力都会有回报。

在算法竞赛中，利用整体和部分的性质可以达成很多目的，例如利用前缀和可以在常数时间复杂度中查询区间和，利用差分在常数时间复杂度对序列进行区间操作，或者利用离散化去除无用数据区间，通过放缩保留有用的数据。这些操作使得我们不必每次都要重复处理个体的数据，而是直接对整体进行处理，从而降低时间复杂度，提升运算效率。

本篇将介绍前缀和的基础算法与实现，帮助读者掌握这一重要的工具。

## 2 前缀和的定义

前缀和是一种简单而高效的算法思想。对于一个数组  $a$ ，它的前缀和数组  $\text{prefix}$  定义如下：

$$\text{prefix}_n = \sum_{i=1}^n a_i$$

通过前缀和，我们可以在  $O(1)$  的时间复杂度内快速计算任意区间  $[l, r]$  的和，公式如下：

$$\text{sum}_{l,r} = \text{prefix}_r - \text{prefix}_{l-1}$$

其中，当  $l = 1$  时， $\text{sum}_{l,r} = \text{prefix}_r$ 。

## 3 算法实现

接下来我们讲上面的思想运用到一道具体的题目中来看一下前缀和的具体使用方法。

由于笔者是洛谷忠实粉丝，这次包括之后的所有练习题可能大概率来自洛谷，建议各位今早注册账号跟着笔者一起练习。

我们选用洛谷的 B3612，由于其实现思路与定义中所讲内容完全雷同，这里直接上代码：

---

```
1 #include <bits/stdc++.h>
3 using namespace std;
5 int n, m, a[100050], s[100050];
7 inline int read() {
    int f = 1, x = 0; char ch = getchar();
9     while (ch < '0' or ch > '9') { if (ch == '-') f = -1; ch = getchar(); }
    while (ch >= '0' and ch <= '9') { x = (x << 3) + (x << 1) + (ch ^ 48); ch =
        ↪ getchar(); }
11    return f*x;
    }
13
14 int main() {
15     n = read();
    for (int i = 1; i <= n; i++)
17         s[i] = s[i - 1] + (a[i] = read()); //构造前缀和数组
    m = read();
19     for (int i = 1; i <= m; i++) {
        int l = read(), r = read();
21         cout << s[r] - s[l - 1] << endl; //根据上文公式得出
    }
23     return 0;
    }
```

目前，如果你能完全理解以上代码，你已经掌握了前缀和的最基础运用。下一章将会介绍前缀和在差分中的运用。

编者按：如果你喜欢比较更偏 C++ 的实现方法，你还可以这样实现前缀和：

```
vector<int> values(n); // 假设其中已有数据
2 vector<int> prefixSums(1);
for (auto i : values) prefixSums.push_back(prefixSums.back() + i);
```

其中 `vector<int> prefixSums(1)` 定义了一个长度为 1 且值为 0 的向量，随后的 for 循环中则每次把 `values` 中的一项与这个前缀和向量的最后一项相加，然后添加到后者中。