

算法竞赛速成

前缀和 差分 & 二分

目录

- 前缀和 差分

 - 什么是前缀和

 - 前缀和的应用

 - 什么是差分

 - 差分的应用

- 二分

 - 二分查找

 - 整数二分

 - 实数二分

前缀和

- 先引出一个问题

现给一个 $1e5$ 的数组，有 q 次询问，每次询问给你一个区间，要求你输出这个区间的所有元素总和

前缀和

- 朴素做法
- 根据每次询问给的区间[l,r]把[l,r]的所有元素都遍历一遍

```
int t;cin >> t;
while(t--){
    int l,r;cin >> l >> r;
    int res = 0;
    for(int i = l;i<=r;i++) res += a[i];
    cout << res << endl;
}
```

但是这个过程的复杂度为 $O(n^2)$,
无法通过 $1e5$ 的数据, 考虑优化

前缀和

- 我们定义数组sum

$$sum_i = \sum_{j=1}^i a_j$$

- 根据此定义有：

$$sum_r - sum_{l-1} = \sum_{i=1}^r a_i - \sum_{i=1}^{l-1} a_i = a_l + a_{l+1} + \dots + a_r$$

- 于是我们就得到了一个可以在 $O(1)$ 时间查询区间和的方法

前缀和

- <https://www.luogu.com.cn/problem/B3612>
- 代码

```
cin >> n;
vector<int> a(n+1),sum(n+1);
for(int i = 1;i<=n;i++) cin >> a[i];
for(int i = 1;i<=n;i++) sum[i] = sum[i-1]+a[i];//预处理sum

int t;cin >> t;
while(t--)
{
    int l,r;cin >> l >> r;
    int res = 0;
    cout << sum[r]-sum[l-1] << endl;
}
```

差分

- 再引出一个问题
- 现给一个 $1e5$ 的数组，有 q 次操作，每次操作给你一个区间，与一个值 x ，要求把这个区间内的所有元素都加上 x
- q 次操作结束后给出一个询问，问第 pos 个元素的值是多少

差分

- 为了解决此类多次区间修改，单次单点查询的问题，我们引入差分数组

$$\therefore \text{minuss}_i = \begin{cases} a_i - a_{i-1} & i \geq 2 \\ a_1 & i = 1 \end{cases}$$

差分

- <https://www.luogu.com.cn/problem/P2367>
- 考虑差分数组的性质
- 手推一下可以发现，把minuss_i加x相当于把a数组中i到n的所有元素都加上x
- 由此我们就得到了一种可以在 $O(1)$ 时间内进行区间修改的数据结构

```
for(int i = 1;i<=r;i++) a[i] += x;  
//等价于  
minuss[1] += x,minuss[r+1] -= x;
```

从一维推广到二维

- 对于二维数组a，我们可以定义a数组的二维前缀和数组sum

$$sum_{i,j} = \sum_{x=1}^i \sum_{y=1}^j a_{x,y}$$

考虑如何求出一个二维数组的前缀和数组

考虑一个具体的例子。

$$A = \begin{pmatrix} 1 & 2 & 4 & 3 \\ 5 & 1 & 2 & 4 \\ 6 & 3 & 5 & 9 \end{pmatrix} \quad S = \begin{pmatrix} 1 & 3 & 7 & 10 \\ 6 & 9 & 15 & 22 \\ 12 & 18 & 29 & 45 \end{pmatrix}$$

这里， S 是给定矩阵 A 的前缀和。根据定义， $S_{3,3}$ 是左图中虚线方框中的子矩阵的和。这里， $S_{3,2}$ 是蓝色子矩阵的和， $S_{2,3}$ 是红色子矩阵的和，它们重叠部分的和是 $S_{2,2}$ 。由此可见，如果直接相加 $S_{3,2}$ 和 $S_{2,3}$ ，会重复计算 $S_{2,2}$ ，所以应该有

$$S_{3,3} = A_{3,3} + S_{2,3} + S_{3,2} - S_{2,2} = 5 + 18 + 15 - 9 = 29.$$

由此得到递推公式

$$sum_{i,j} = a_{i,j} + sum_{i-1,j} + sum_{i,j-1} - sum_{i-1,j-1}$$

```
cin >> n >> m;
vector<vector<int>> a(n+1,vector<int>(m+1)),sum(n+1,vector<int>(m+1));
for(int i = 1;i<=n;i++)
{
    for(int j = 1;j<=m;j++) cin >> a[i][j];
}

for(int i = 1;i<=n;i++)
{
    for(int j = 1;j<=m;j++) sum[i][j] = sum[i-1][j]+sum[i][j-1]+a[i][j]-sum[i-1][j-1];
}
```

再再引出一个问题

- 给定一个矩阵a，左上点(x1,y1)，右下点(x2,y2)，我们如何利用前缀和数组查询这两个点围成的子矩阵的元素和呢？

同样的道理，在已经预处理出二位前缀和后，要查询左上角为 (i_1, j_1) 、右下角为 (i_2, j_2) 的子矩阵的和，可以计算

$$S_{i_2, j_2} - S_{i_1, j_2} - S_{i_2, j_1} + S_{i_1, j_1}.$$

二维前缀和例题

- <https://www.luogu.com.cn/problem/P1387>

题目描述

 复制Markdown  展开

在一个 $n \times m$ 的只包含 0 和 1 的矩阵里找出一个不包含 0 的最大正方形，输出边长。

输入格式

输入文件第一行为两个整数 $n, m (1 \leq n, m \leq 100)$ ，接下来 n 行，每行 m 个数字，用空格隔开，0 或 1。

输出格式

一个整数，最大正方形的边长。

二维差分

- <https://www.cnblogs.com/csai-H/p/16931333.html>

```
/*
    将以(x1,y1)为左上点,(x2,y2)为右下点的矩阵中
    所有元素+1
*/
int x1,y1,x2,y2;
cin >> x1 >> y1 >> x2 >> y2;
a[x1][y1] += 1;
a[x1+1][y2] -= 1;
a[x2][y1+1] -= 1;
a[x2+1][y2+1] += 1;
```

二分

- 二分查找

对于一个有序数组a， 我们可以在 $O(\log n)$ 的时间内查找这个元素

P.S.在STL中提供了查找首个不小于定值x的元素的函数`std::lowerbound`与查找首个大于定值x的元素的函数`std::upperbound`， 两个函数都基于二分查找实现

二分

- 引入
- 游戏规则:
 - 系统会随机生成一个1到1024的数 x
 - 每次你可以输入一个数
 - 系统会告诉你，你输入的数是偏大还是偏小，又或者你猜对了
 - 你只有11次机会

二分

- 不难发现，我们每次猜数都能根据已知的信息缩小 x 的范围。
- 我们令 l 为 x 最小可能是多少， r 为 x 最大可能是多少
- 当我们猜一个数 y ， x 的范围会被缩小到 $l \sim y-1$ 或者 $y+1 \sim r$
- 如果我们取 $y=(l+r)/2$ ，那么每次猜数都可以将范围减半
- $\log_2 1024=10$ 次后，范围大小就会变为1
- 所以11次一定能把数猜出来

二分

- 二分实际上分的是答案，分出答案后将这个答案代入检验是否合法

```
4  int n;
5  vector<int> a = {0,1,3,4,5,7,8,10,12};
6
7  bool check(int x,int tar)
8  {
9      if(a[x]>=tar) return 1;
10     else return 0;
11 }
12
13 void solve()
14 {
15     n = a.size()-1;
16     int tar = 6;
17     int l = 1,r = n;//r,l是答案的上界与下界
18     while(l<r)
19     {
20         int mid = l+r >> 1;
21         if(check(mid,tar)) r = mid;//此时mid可能作为我们的答案，所以取等
22         else l = mid+1;//此时l不可能作为我们的答案，不取等
23     }
24     cout << l << endl;
25 }
26
```

二分

- 如果你要使用二分算法，那么必须具备的条件是
 - 单调性
 - check函数可写

<https://www.luogu.com.cn/problem/P2678>

题目描述

一年一度的“跳石头”比赛又要开始了！

这项比赛将在一条笔直的河道中进行，河道中分布着一些巨大岩石。组委会已经选择好了两块岩石作为比赛起点和终点。在起点和终点之间，有 N 块岩石（不含起点和终点的岩石）。在比赛过程中，选手们将从起点出发，每一步跳向相邻的岩石，直至到达终点。

为了提高比赛难度，组委会计划移走一些岩石，使得选手们在比赛过程中的最短跳跃距离尽可能长。由于预算限制，组委会至多从起点和终点之间移走 M 块岩石（不能移走起点和终点的岩石）。

实数二分

- 理论上二分是可以无限分割的，我们之前学习的整数二分相当于把二分的精度定在了“整数”
- 而实数二分的思路和整数二分几乎一样，只需要多引入一个参数来确定二分的精度即可

二分

- 实数二分
- <https://www.luogu.com.cn/problem/P1024>

题目描述

 复制Markdown  展开

有形如： $ax^3 + bx^2 + cx + d = 0$ 这样的三个一元三次方程。给出该方程中各项的系数 (a, b, c, d 均为实数)，并约定该方程存在三个不同实根（根的范围在 -100 至 100 之间），且根与根之差的绝对值 ≥ 1 。要求由小到大依次在同一行输出这三个实根(根与根之间留有空格)，并精确到小数点后 2 位。

提示：记方程 $f(x) = 0$ ，若存在 2 个数 x_1 和 x_2 ，且 $x_1 < x_2$ ， $f(x_1) \times f(x_2) < 0$ ，则在 (x_1, x_2) 之间一定有一个根。