

CDQ分治

📅 2020-11-19 | 📅 2021-01-15 | 📁 OI & ACM | 👁 5

📄 1.6k | ⌚ 1 分钟

简介

大概是一种特殊的分治吧

一般可以离线做的题目都可以尝试用 *cdq* 来解决时间这个维度的问题

我们来看一个比较经典的例子，单点加，区间求和

首先将区间求和拆成两个前缀

我们将操作进行的时间看成一个维度，另一个查询或者是更改的位置看成第二个维度

```
1  #include <iostream>
2  #include <cstdio>
3  #include <algorithm>
4  #define cQ const Query
5  #define maxn 500010
6  using namespace std;
7
8  int n, m;
9
10 struct Query {
11     int x, v, id;
12
13     Query() {}
14     Query(int _x, int _v, int _id) { x = _x; v = _v; id = _id; }
15
16     friend bool operator < (cQ &u, cQ &v) { return u.x < v.x; }
17 } Q[maxn * 3]; int cnt;
18
19 int ans[maxn], cans;
20 void cdq(int l, int r) {
```

```

21     if (l == r) return ; int m = l + r >> 1, j = l - 1, sum = 0;
22     cdq(l, m); cdq(m + 1, r);
23     for (int i = m + 1; i <= r; ++i) {
24         int x = Q[i].x, v = Q[i].v, id = Q[i].id;
25         if (!id) continue;
26         while (j < m && Q[j + 1].x <= x) {
27             ++j; if (Q[j].id) continue;
28             sum += Q[j].v;
29         } ans[id] += v * sum;
30     }
31     inplace_merge(Q + l, Q + m + 1, Q + r + 1);
32 }
33
34 int main() {
35     cin >> n >> m;
36     for (int i = 1; i <= n; ++i) {
37         int x; scanf("%d", &x);
38         Q[++cnt] = Query(i, x, 0);
39     }
40     for (int i = 1; i <= m; ++i) {
41         int opt, x, y; scanf("%d%d%d", &opt, &x, &y);
42         if (opt == 1) Q[++cnt] = Query(x, y, 0);
43         else {
44             Q[++cnt] = Query(x - 1, -1, ++cans);
45             Q[++cnt] = Query(y, 1, cans);
46         }
47     }
48     cdq(1, cnt);
49     for (int i = 1; i <= cans; ++i) printf("%d\n", ans[i]);
50     return 0;
51 }

```

应用

1. 偏序问题

cdq 分治可以自动解决一维，剩下的维度可以用其它数据结构来做

当然也可以使用 *cdq* 套 *cdq* 来解决两维

[Luogu【模板】三维偏序（陌上花开）](#)

[Luogu P3769 \[CH弱省胡策R2\]TATT 四维偏序](#)

2. 某些带修数据结构题

当修改可以分批次贡献的时候，我们就可以考虑使用 *cdq* 分治来将操作离线下来

技巧

1. 分治后归并

大概就是利用两个儿子的信息

[Luogu P3374 【模板】树状数组 1\(CDQ分治\)](#) 这道题如果分治后归并的话，复杂度是 $O(n \log^2 n)$ 的

2. *dp* 类题目在 *cdq* 分治时可能要向左递归，计算跨越中点的贡献，再向右递归

[Luogu P3769 \[CH弱省胡策R2\]TATT](#)

本文结束  感谢阅读

[# Tech](#) [# CDQ分治](#)

< [CF 1447F2 Frequency Problem \(Hard Version\)](#)

[Luogu P3374 【模板】树状数组 1\(CDQ分治\)](#) >

© 2020 – 2022  DDOSvoid

 1.8m |  27:07

9089 |  17849

由 [Hexo](#) & [NexT.Gemini](#) 强力驱动