知识精炼(五)

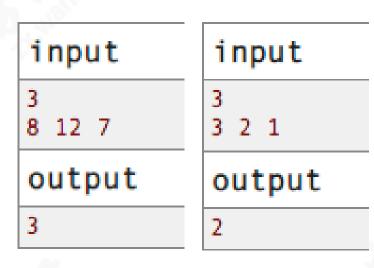
全 主讲人:邓哲也



给定一个数列 A[1], A[2], …, A[n].

问有多少个有序对(x, y),满足:

$$x < y$$
 $A[x] >= y$
 $A[y] >= x$
 $n <= 200000$



看上去要满足三个条件,实际上我们可以合并前两个得到:

统计 x < y <= A[x] 且 A[y] >= x 的数对。

这类问题,我们一般通过对一维排序,按顺序处理,来满足 这维要求。

然后统计第二维的信息即可。

我们从大到小枚举 x。

然后把 A[y] >= x 的 y 记录下来。

然后查询在区间(x, A[x]]中的 y 的个数。

显然用树状数组就可以做到。

时间复杂度 0(n log n).

一个 trick 是如果 A[i] > n, 那么可以把 A[i] 改为 n,

因为大于 n 和等于 n 的效果是相同的,这样可以避免离散

化。

```
#define N 200010
int ord[N], n, a[N], bit[N];
void add(int x, int d) {
    for (;x \le n;x += x \& -x) bit[x] += d:
int ask(int x) {
    int ret = 0;
    for (;x;x -= x \& -x) ret += bit[x];
    return ret;
```

```
int cmp(int i, int j) {
    return a[i] > a[j];
int main() {
    scanf("%d", &n);
    for (int i = 1; i \le n; i ++) {
        scanf("%d", &a[i]);
        a[i] = min(a[i], n);
        ord[i] = i;
```

```
sort(ord + 1, ord + n + 1, cmp);
int j = 0;
long long ans = 0;
for (int i = n; i >= 1; i --) {
    while (j + 1 \le n \&\& a[ord[j + 1]] \ge i) {
        add(ord[j + 1], 1);
        j ++;
    if (i \le a[i])
       ans += ask(a[i]) - ask(i);
cout << ans << endl;
return 0;
```

下节课再见