

2020.08.06codeforces加训

比赛情况

题号	A	B	C	D	E	F	G	H	I	J
状态	O	-	O	-	-	-	O	Ø	O	O

O 在比赛中通过 Ø 赛后通过! 尝试了但是失败了- 没有尝试

比赛时间

2020-08-06 12:00-17:00

题解

A - Hacker Cups and Balls

题意：

给定一个序列，给q次操作，每次操作升序或降序排列某区间中的数，问q次操作后序列中间位置的数是什么。

题解：

我们可以考虑二分答案，然后把所有大于当前二分出答案的数字赋值为1，其余的赋值为0，这样一次排序操作其实就是查询一个区间有多少个0多少个1，然后把前后两部分分别连续赋值为0和1，这样一个操作通过线段树就能完成。

然后查询中间位置的数字是0还是1，如果是1那么中位数就应该是更大，如果为0那么中位数就可能是当前答案或更小。

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e5+5;
int tr[N<<2], lazy[N<<2], a[N];
struct Node {
    int l, r;
} q[N];
int midans, n, m;
void Push_Down(int p, int l, int r) {
    if (lazy[p] == -1 || l == r) return;
    int mid = (l+r)>>1;
    tr[p<<1] = (mid-l+1)*lazy[p];
    tr[p<<1|1] = (r-mid)*lazy[p];
    lazy[p<<1] = lazy[p<<1|1] = lazy[p];
    lazy[p] = -1;
}
void Build(int p, int l, int r) {
```

```
lazy[p] = -1;
if (l==r) {
    tr[p] = a[l] > midans;
    return;
}
int mid = (l+r)>>1;
Build(p<<1,l,mid);
Build(p<<1|1,mid+1,r);
tr[p] = tr[p<<1]+tr[p<<1|1];
}

void Update(int p,int l,int r,int a,int b,int c) {
    if (a > b) return;
    Push_Down(p,l,r);
    if (l >= a && r <= b) {
        tr[p] = (r-l+1)*c;
        lazy[p] = c;
        return;
    }
    int mid = (l+r)>>1;
    if (a <= mid) Update(p<<1,l,mid,a,b,c);
    if (b > mid) Update(p<<1|1,mid+1,r,a,b,c);
    tr[p] = tr[p<<1] + tr[p<<1|1];
}

int Getans(int p,int l,int r,int a,int b) {
    Push_Down(p,l,r);
    if (l >= a && r <= b) return tr[p];
    int ans = 0;
    int mid = (l+r)>>1;
    if (a <= mid) ans+= Getans(p<<1,l,mid,a,b);
    if (b > mid) ans+= Getans(p<<1|1,mid+1,r,a,b);
    return ans;
}

bool check(int x) {
    midans = x;
    Build(1,1,n);
    for (int i = 1;i<= m;i++) {
        int L = min(q[i].l,q[i].r);
        int R = max(q[i].l,q[i].r);
        int tmp = Getans(1,1,n,L,R);
        if (q[i].l < q[i].r) {
            Update(1,1,n,L,R-tmp,0);
            Update(1,1,n,R-tmp+1,R,1);
        } else {
            Update(1,1,n,L,L+tmp-1,1);
            Update(1,1,n,L+tmp,R,0);
        }
    }
    int midpos = (n>>1)+1;
    int t = Getans(1,1,n,midpos,midpos);
    return t == 1;
}
```

```

}
int GetFinalans() {
    int l = 1, r = n+1;
    while (l < r) {
        int mid = (l+r)>>1;
        if (check(mid)) l = mid+1;
        else r = mid;
    }
    return l;
}
int main()
{
    scanf("%d%d", &n, &m);
    for (int i = 1; i <= n; i++) scanf("%d", &a[i]);
    for (int i = 1; i <= m; i++)
        scanf("%d%d", &q[i].l, &q[i].r);
    printf("%d\n", GetFinalans());
    return 0;
}

```

J - Zero Game

题意：

给定一个01序列，定义一次操作是把任意一个字符换到任意一个位置，有 q 次询问，第 i 次询问问 k_i 次操作后这个序列最长可以有多长的连续0

题解：

考虑只对于一个区间内的字符进行操作的答案，因为有两种可能，一种是把1换走使得两个连续的0序列合并，另一种是换进来一些0，那么我们可以发现对于一个区间 (l, r) 如果 $sum_r - sum_{l-1} \leq k_i$ ，那么这个区间的答案应该是 $l+r-1-(sum_r - sum_{l-1}) + (k_i - (sum_r - sum_{l-1}))$ （ sum_i 为从位置 1 到 i 之间有多少个1），那么整理一下，这个式子变成了 $r - 2 \times sum_r + 2 \times sum_{l-1} - (l-1) + k_i$ ，对于同一个 r 我们只需要找一个满足 $sum_r - sum_{l-1}$ 的 l 中 $2 \times sum_{l-1} - (l-1)$ 最大的即可，一开始觉得是数据结构，后来发现单调队列维护就行了。

```

#include <bits/stdc++.h>
using namespace std;
const int N = 1e6+5;
char s[N];
int sum[N];
int lsum[N];
int k[N], id[N], ans[N], que[N], head, tail;
int main()
{
    int n, q;
    scanf("%s", s+1);
    n = strlen(s+1);

```

```
scanf("%d",&q);
for (int i = 1;i<= q;i++)
{
    scanf("%d",&k[i]);
    id[i] = k[i];
}
sort(id+1,id+q+1);
int cnt = 0;
for (int i = 1;i<= q;i++)
    if (id[i]!=id[cnt])
        id[++cnt] = id[i];
for (int i = 1;i<= n;i++)
{
    sum[i] = sum[i-1]+(s[i]=='1');
    lsum[i] = 2*sum[i]-i;
}
for (int i = 1;i<= cnt;i++)
{
    int K = id[i];
    head = tail = 1;
    que[1] = 0;
    ans[K] = ans[id[i-1]];
    for (int j = 1;j<= n;j++) {
        while (head <= tail && sum[j]-sum[que[head]] > K) head++;
        if (head<=tail)ans[K] =
max(ans[K],lsum[que[head]]+j-2*sum[j]+K);
        while (head <= tail && lsum[j]>=lsum[que[tail]])tail--;
        que[++tail] = j;
    }
    ans[K] = min(ans[K],n-sum[n]);
}
for (int i = 1;i<= q;i++)
    printf("%d\n",ans[k[i]]);
return 0;
}
```

比赛总结与反思

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:20200806%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95&rev=1596723626

Last update: 2020/08/06 22:20