

FOI2023 C班 DAY1简易题解

- T1:map简单应用
- T2:单调栈
- T3:单调队列

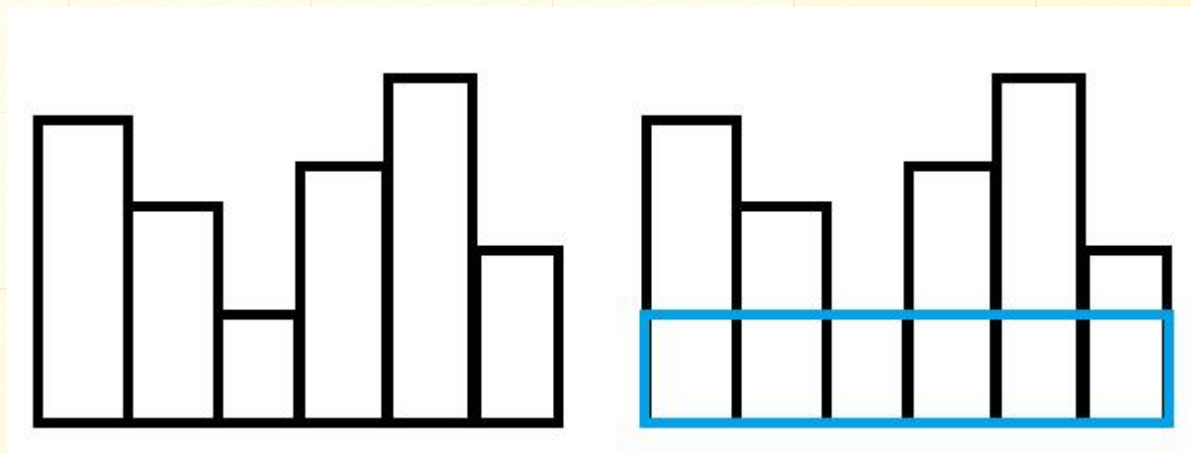
T1:map简单应用

- 用map建立：广告名和收入对应的映射
- `map<string,double> val;`

```
1 //代码如下:
2     map<string,double> val;
3     scanf("%d%d%d\n",&n,&m,&k);
4     for (int i=1;i<=n;++i)
5     {
6         cin>>s1;
7         scanf("%lf\n",&v);
8         val[s1]=v;
9     }
10    for (int i=1;i<=m;++i)
11    {
12        cin>>s1>>s2;ans+=val[s2];
13    }
14    value=ans;
15    num=value/k;
16    cout<<num<<endl;
17
```

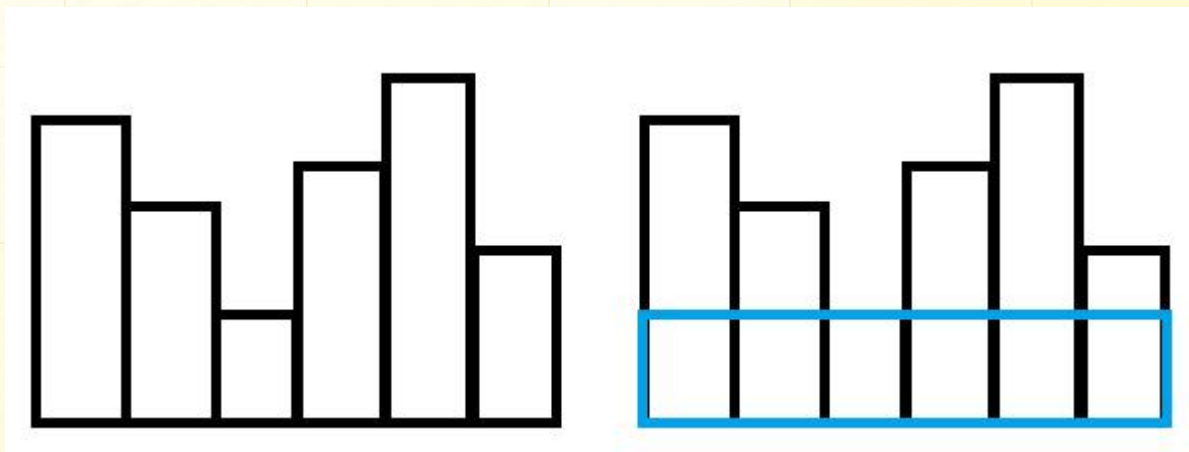
T2:矩形墙

- 穷举法时间复杂度

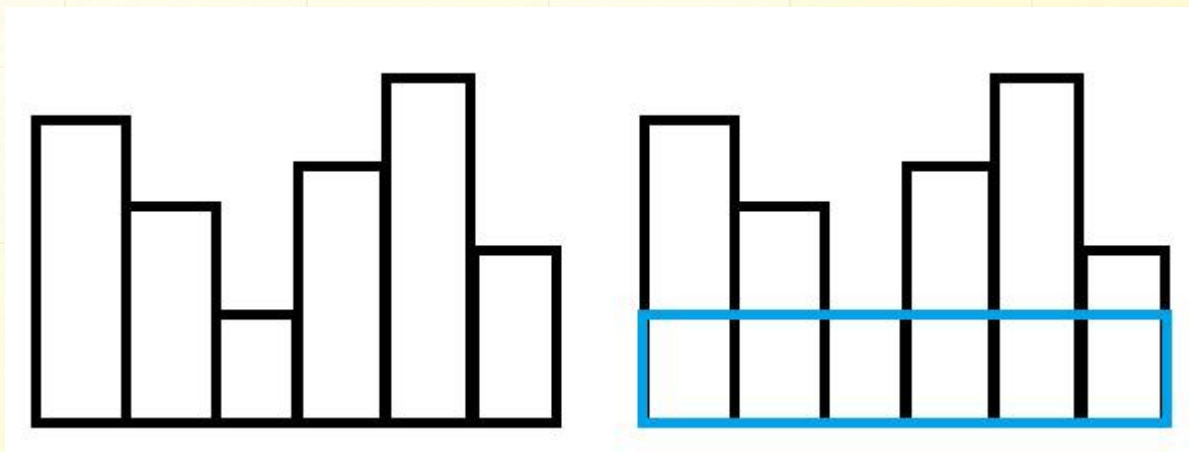


T2:矩形墙

- 思路
- 穷举每一个 $h[i]$,以 $h[i]$ 的高度为矩形墙的高度, 计算对应的矩形墙的宽度?



- 用一个单调递增栈记录每个矩形向左和向右延伸的最远距离(矩形墙的所有高度为 $h[i]$, 矩形墙的其余单位矩形的高度 $\geq h[i]$)



- 当前节点的坐标为 i ，高度为 $h[i]$ ，从当前点开始，向左找：第一个比当前节点高度小的坐标，维护一个单调（？）的栈

- 当前节点的坐标为 i ，高度为 $h[i]$ ，从当前点开始，向左找：第一个比当前节点高度小的坐标 $l[i]$ ，维护一个单调（递增）的栈？
- 当前节点的坐标为 i ，高度为 $h[i]$ ，从当前点开始，向右找：第一个比当前节点高度小的坐标 $r[i]$ ，维护一个单调（递增）的栈？

```
1 top=0;
2 for(int i=1;i<=n;++i)
3 {
4     //找左边第一个小于h[i]的坐标
5     scanf("%d",&h[i]);
6     while(top>0&& h[st[top]]>=h[i]) --top;
7     if (top==0) l[i]=0;else l[i]=st[top];
8     st[++top]=i;    //st[]存储坐标
9 }
10 top=0;
11 for(int i=n;i>=1;--i)
12 {
13     //找右边第一个小于h[i]的坐标
14     while(top>0&&h[st[top]]>=h[i]) --top;
15     if (top==0) r[i]=n+1;else r[i]=st[top];
16     st[++top]=i;
17 }
18 long long ans=0;
19 for(int i=1;i<=n;++i)
20 ans=max(ans,(long long)h[i]*(r[i]-l[i]-1));
21 printf("%lld\n",ans);
```

T3:万人狂欢节

- 6 4
- 10 3
- 6 2
- 5 3
- 9 7
- 3 6
- 11 2

- 穷举法时间复杂度？

这道题是典型的单调队列的题。

思路：

遍历每个人，查找它前后 \varnothing 距离以内的颜值 \geq 自身颜值2倍的人。这里做一个推导：如果一个人左右 \varnothing 范围以内最高颜值都没有 \geq 自身颜值的2倍，那他一定不是low-man。

所以，分别找每个人前后 \varnothing 距离内**最高颜值**，判断是否 \geq 自身颜值的2倍，如果左右都有，low-man人数+1。

用一个单调递减队列存每个人，每个人入队时把它前面颜值低于它的人都踢出，还要把队首所有坐标不在距离 \varnothing 范围以内的人出队，最后判断队首的颜值是否 \geq 自身颜值的2倍即可。

这样遍历前、后各一次，两次都满足条件low-man人数增加1。

T3:单调队列

```
1 //从左往右遍历(从右往左类似)代码如下:
2     n=read(),m=read();
3     for(i=1;i<=n;i++)man[i].x=read(),man[i].c=read();
4     sort(man+1,man+1+n,cmp);
5  □ for(i=1;i<=n;i++){ // l队头 r队尾
6         while(r>l&&man[f[l]].x<man[i].x-m) l++;
7         if(r>l&&man[f[l]].c>=man[i].c*2) vl[i]=1;
8         while(r>l&&man[f[r-1]].c<man[i].c) r--;
9         f[r++]=i; //f[]存下标
10    }
```

The background features a light yellow grid on the left side, with a white border separating it from the right side. On the right side, there are large, overlapping, curved shapes in red, yellow, green, and blue, separated by white borders. The word "END" is written in a bold, orange-red font on the left side, overlapping the grid.

END