

# 题解

# 目录

#D、积木城堡

## #D、积木城堡

### 题目大意

在一个长为  $K$  宽为  $L$  的桌面上，有一个积木城堡。已知城堡的主视图与左视图，求出积木个数的最大值与最小值。若无解，输出-1。

## #D、积木城堡

### 思路

要先判断是否有合法输出，若在主视图中的最高值与左视图中的最高值不相等，就没有合法方案，输出-1。设城堡最高层记为maxx，将1到maxx层中，把主视图能看见的每层的积木个数记录进c数组，把左视图能看见的每层的积木个数记录进d数组，然后求最小值和最大值。

## #D、积木城堡

最小值

因为要求最小值，所以要让从主视图看到的和从左视图看到的高度相同的列变成同一列，所以要枚举每个高度，加上 $\min(c[i], d[i])$ 。

## #D、积木城堡

最大值

设主视图中第 $i$ 列的高度为 $a[i]$ ,左视图中第 $j$ 列的高度为 $b[j]$ 。需要让每一列上都尽量高，所以第 $i$ 行第 $j$ 列的高度要是 $\max(a[i], b[j])$ 。枚举每一列求出最大值，时间复杂度是 $O(k \cdot l)$ ，能拿到40分。

将主视图和左视图排序，枚举主视图上的每一列，用前缀和求出每一列上的最大值，求和，就求出最大值了。时间复杂度是 $O(k \log k + l \log l)$

# #D、积木城堡

最小值

最大值

1	1			
4		3	4	
2				2
	1	3	4	2

1	1	1	1	1
4	1	3	4	2
2	1	2	2	2
	1	3	4	2

# #D、积木城堡

## 核心代码

```
5 int main(){
6     cin >> n >> m;
7     for(ll i = 1; i <= n; i++){
8         cin >> a[i];
9         c[a[i]]++;
10        ma = max(ma, a[i]);
11    } sort(a + 1, a + n + 1);
12    for(ll i = 1; i <= m; i++){
13        cin >> b[i];
14        d[b[i]]++;
15        mb = max(mb, b[i]);
16    } sort(b + 1, b + m + 1);
17    // 无解
18    if(ma != mb){
19        cout << "-1";
20        return 0;
21    } maxx = max(ma, mb);
```

```
22 // 最小
23 for(ll i = 0; i <= maxx; i++)
24     mi += max(c[i], d[i]) * i;
25 // 最大
26 b[m + 1] = INT_MAX;
27 for(ll i = 1; i <= n; i++)
28     e[i] = e[i - 1] + a[i];
29 for(ll i = 1; i <= m; i++)
30     f[i] = f[i - 1] + b[i];
31 for(ll i = 1; i <= n; i++){
32     while(j <= m + 1 && b[j] <= a[i]) j++;
33     mx += f[j - 1] + (m - j + 1) * a[i];
34 }
35 cout << mi << " " << mx;
36 return 0;
37 }
38
```