

A 题解题报告

描述：

N 个怪兽，有两种属性 s , d 。 s 是每个怪兽的能力值， d 是代表他希望拿到一个编号 1 到 d 的盒子。我们能够安排他们选择的顺序，如果他们拿不到自己想要的盒子，我们就会收到 s 的伤害。要我们求出最小伤害。

思路：

贪心。

我们很容易想到如果两个怪兽如果发生冲突，我们当然愿意满足能力值比较高的怪兽。我们安排怪兽时也希望他拿到的盒子编号尽量靠后，因为前面的盒子可能适合更多的怪兽。

有了这两点基本思路，基本这个题就解决了。把怪兽按照 s 降序排序。从前往后安排怪兽，每次安排怪兽都安排距离他的 d 的最近的盒子。

但是如何确定 d 最近的那个盒子呢？

其实这里和链表比较相似。一个盒子连接着前一个盒子，如果一个盒子被选走，那么我们就把这个盒子删掉。每次选择当前盒子的下一个盒子。这里我就想到我们当时做国外区域赛的一道题目，**orz** 陈家泽大神给出的并查集思想。

每个盒子看成一个点，每个点初始是在一个集合中。每次我们删掉这个点，就要把当前点集合和前一个点集合合并，这样我们求一次祖先就知道距离当前 d 最近的那个盒子编号了。

Code:

```
#include <cstring>
#include <iostream>
#include <cstdlib>
#include <cstdio>
#include <algorithm>
using namespace std;
#define N 50050
struct node
{
    int d,s;
}a[N];
int vis[N],set[N];
int cmp(node a,node b)
{
    return a.s>b.s;
}
int findset(int x)
{
    if (set[x]==x)
        return x;
    return (set[x]=findset(set[x]));
}
void Union(int x,int y)
```

```

{
    int setx=findset(x);
    int sety=findset(y);
    if (setx!=sety)
        set[setx]=sety;
}
int main (void)
{
    int n,m;
    while (cin>>n>>m && (n||m))
    {
        memset(vis,0,sizeof(vis));
        for (int i=1;i<=m;i++)
            set[i]=i;
        for (int i=1;i<=n;i++)
            scanf("%d",&a[i].d);
        for (int i=1;i<=n;i++)
            scanf("%d",&a[i].s);
        sort(a+1,a+1+n,cmp);
        int ans=0,now;
        for (int i=1;i<=n;i++)
        {
            now=findset(a[i].d);
            if (vis[now]==0)
                vis[now]=1;
            else ans+=a[i].s;
            if (now>1)
                Union(now,now-1);
        }
        cout<<ans<<endl;
    }
    return 0;
}

```