约会大作战-青春版

题目描述

学校正在举办联谊活动！目前报名的有 n 个男生和 n 个女生。其中第 i 个男生魅力值为 ai，只喜欢魅力值不小于 pi 的女生；第 i 个女生魅力值为 bi，只喜欢魅力值不小于 qi 的男生。

男生与女生能约会当且仅当两人相互喜欢。作为一个没有钞能力的主办方，the_ignorant 只能希望让尽可能多的人能约会成功，那么最多有多少对情侣能约会成功呢?

输入格式

第一行一个正整数 n（1≤n≤4001≤n≤400），表示男生和女生的数量。

第二行 n 个非负整数 a1,a2,…,an（0≤ai≤10^9），表示男生的魅力值。

第三行 n 个非负整数 p1,p2,…,pn（0≤pi≤10^9），表示男生喜欢的女生魅力值最小值。

第四行 n 个非负整数 b1,b2,…,bn（0≤bi≤10^9），表示女生的魅力值。

第五行 n 个非负整数 q1,q2,…,qn（0≤qi≤10^9），表示女生喜欢的男生魅力值最小值。

输出格式

一行一个非负整数，表示最多能有几对情侣约会成功。

二分图匹配

二分图：

二分图又称作二部图，是图论中的一种特殊模型。 设G=(V,E)是一个无向图，如果顶点V可分割为两个互不相交的子集(A,B)，并且图中的每条边（i，j）所关联的两个顶点i和j分别属于这两个不同的顶点集(i in A,j in B)，则称图G为一个二分图。

匈牙利算法：

匈牙利算法是由匈牙利数学家 `Edmonds` 于1965年提出，因而得名。匈牙利算法是基于 `Hall` 定理中充分性证明的思想，它是部图匹配最常见的算法，该算法的核心就是寻找增广路径，它是一种用增广路径求二分图最大匹配的算法。

题目分析

- 该题是一个二分图最大匹配的问题，可以采用匈牙利算法解决
- 可以将男生集合和女生集合共同看成一个二分图，满足约会成功条件的男生和女生之间存在边，让尽可能多的人约会成功即求这个二分图的最大匹配
- 用一个结构体记录每个人的魅力值和达到喜欢的最小魅力值，两个数组分别记录男生和女生

```
typedef struct{
    int a,p;
} NODE;
NODE boy[N],girl[N];
```

约会成功的判断标准则为

```
girl[i].a>=boy[j].p && boy[j].a>=girl[i].p
```

- 算法的核心为分别从图一中的每一个点出发，遍历图二中相邻的点，若没有匹配点，或有匹配点但匹配点还有其他选择，则更改匹配

```
bool found(int x)
{
    for(int i=1; i<=k; i++)
    {
        if(line[x][i]&&!used[i])
        {
            used[i]=1;
            if(result[i]==0||found(result[i]))
            {
                result[i]=x;
                return 1;
            }
        }
    }
    return 0;
}
```

代码

```
#include<cstdio>
#include<cstring>
#include<iostream>
#include<queue>
#include<vector>
#include<cmath>
#include<algorithm>
using namespace std;
const int N=405;
int line[N][N];
int result[N],used[N];
int k;
typedef struct{
    int a,p;
} NODE;
NODE boy[N],girl[N];
bool found(int x)
{
    for(int i=1; i<=k; i++)
    {
        if(line[x][i]&&!used[i])
        {
```

```
            used[i]=1;
            if(result[i]==0||found(result[i]))
            {
                result[i]=x;
                return 1;
            }
        }
    }
    return 0;
}
int main()
{
    int i,j;
    scanf("%d",&k);
    memset(line,0,sizeof(line));
    memset(result,0,sizeof(result));
    for (i=1;i<=k;i++)
        scanf("%d",&boy[i].a);
    for (i=1;i<=k;i++)
        scanf("%d",&boy[i].p);
    for (i=1;i<=k;i++)
        scanf("%d",&girl[i].a);
    for (i=1;i<=k;i++)
    {
        scanf("%d",&girl[i].p);
        for (j=1;j<=k;j++)
        {
            if (girl[i].a>=boy[j].p && boy[j].a>=girl[i].p)
                line[j][i]=1;
        }
    }
    int sum=0;
    for(i=1; i<=k; i++)
    {
        memset(used,0,sizeof(used));
        if(found(i)) sum++;
    }
    printf("%d\n",sum);
    return 0;
}
```