E5-C 约会大作战-青春版

20374126 王奘堃

目录

CONTENTS

题面

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。

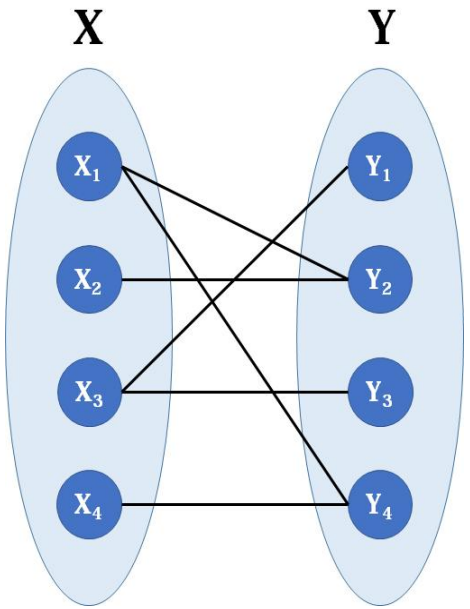
男生与女生能约会当且仅当两人相互喜欢。作为一个没有钞能力的主办方，the_ignorant 只能希望让尽可能多的人能约会成功，那么最多有多少对情侣能约会成功呢?

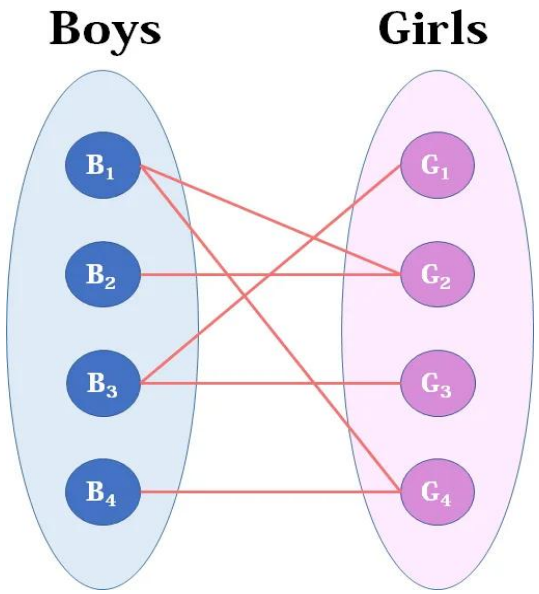
输入格式:
第一行一个正整数 n（1≤n≤400），表示男生和女生的数量。
接下来四行分别输入n个非负整数表示男生的魅力值、男生喜欢的女生魅力值最小值、女生的魅力值、女生喜欢的男生魅力值最小值。

匈牙利算法

Hungarian algorithm

匈牙利算法（Hungarian algorithm）。匈牙利算法主要用于解决一些与二分图匹配有关的问题。
二分图（Bipartite graph）是一类特殊的图，它可以被划分为两个部分，每个部分内的点互不相连。下图是典型的二分图。





在上面的二分图中，每条边的端点都分别处于点集X和Y中。匈牙利算法可以解决二分图的最大匹配数问题。

匈牙利算法

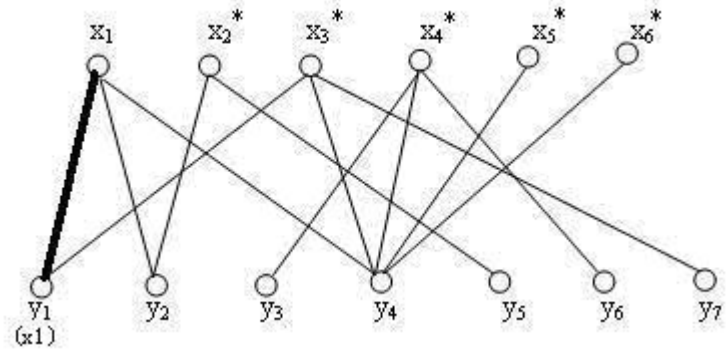
Hungarian algorithm

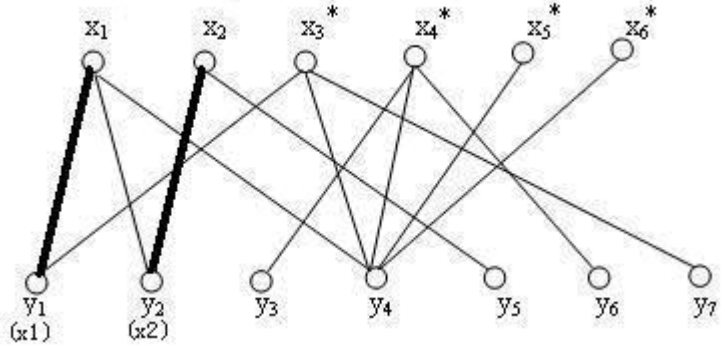
关键步骤：寻找增广路径

交错路径：给定图G的一个匹配M，如果一条路径的边交替出现在M中和不出现在M中，我们称之为一条M-交错路径。

而如果一条M-交错路径，它的两个端点都不与M中的边关联，我们称这条路径叫做M-增广路径。

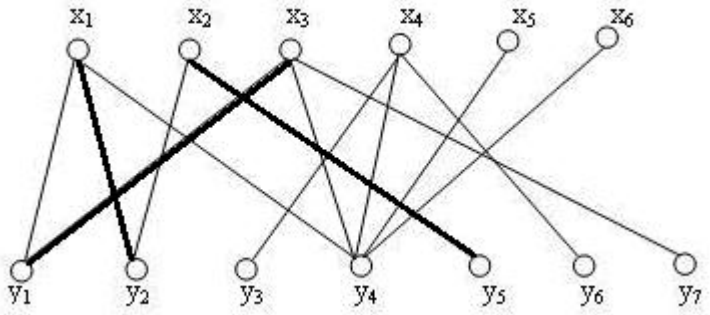
刚开始，一个匹配都没有，我们随意选取一条边，（x1, y1）这条边，构建最初的匹配

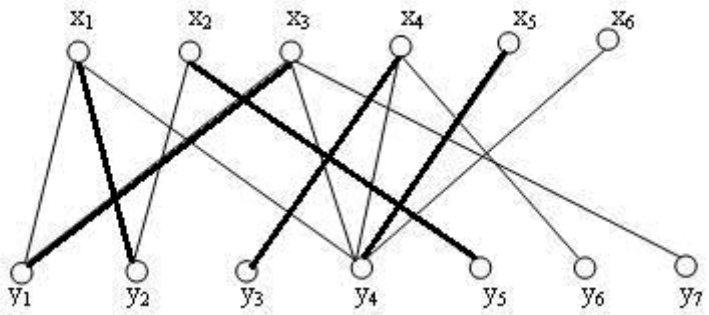




匈牙利算法

Hungarian algorithm





1. 匈牙利算法寻找最大匹配，就是通过不断寻找原有匹配M的增广路径，因为找到一条M匹配的增广路径，就意味着一个更大的匹配M'，其恰好比M 多一条边。

2. 对于图来说，最大匹配不是唯一的，但是最大匹配的大小是唯一的。

代码

核心代码：

```cpp
int p[MAXN];          //记录当前右侧元素所对应的左侧元素
bool vis[MAXN];       //记录右侧元素是否已被访问过
bool match(int i)
{
    for (int j = 1; j <= N; ++j)
        if (Map[i][j] && !vis[j]) //有边且未访问
        {
            vis[j] = true;                  //记录状态为访问过
            if (p[j] == 0 || match(p[j])) //如果暂无匹配，或者原来匹配的左侧元素可以找到新
            {
                p[j] = i;     //当前左侧元素成为当前右侧元素的新匹配
                return true; //返回匹配成功
            }
        }
    return false; //循环结束，仍未找到匹配，返回匹配失败
}
int Hungarian()
{
    int cnt = 0;
    for (int i = 1; i <= M; ++i)
    {
        memset(vis, 0, sizeof(vis)); //重置vis数组
        if (match(i))
            cnt++;
    }
    return cnt;
```

E5-C-Hungary.cpp

E5-C-Hungary.cpp > main()

```cpp
{
    int destination;
    edge *next;
} e[400 * 400];
int cnt;
edge *u[405];
int vis[405], match[405];
int res = 0;
bool Hungarian(int x)···
int main()
{
    int n;
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) scanf("%d", &b[i].charm);
    for (int i = 1; i <= n; i++) scanf("%d", &b[i].require);
    for (int i = 1; i <= n; i++) scanf("%d", &g[i].charm);
    for (int i = 1; i <= n; i++) scanf("%d", &g[i].require);
    for (int i = 1; i <= n; i++) u[i] = NULL;
    for (int i = 1; i <= n; i++){
      for (int j = 1; j <= n; j++){
        if (b[i].charm >= g[j].require && b[i].require <= g[j].charm){
          e[++cnt].destination = j, e[cnt].next = NULL;
          if (u[i] == NULL) u[i] = &e[cnt];
          else e[cnt].next = u[i],u[i] = &e[cnt];
        }
      }
    }
    for (int i = 1; i <= n; i++){
      for (int j = 1; j <= n; j++) vis[j] = 0;
      if (Hungarian(i)) res++;
    }
    printf("%d", res);
    return 0;
}
```

感谢您的耐心观看

20374126 王奘堃