

## 专题4

### 目录

Problem ID	Title
1001	<a href="#">Machine Schedule</a>
1002	<a href="#">Air Raid</a>
1003	<a href="#">棋盘游戏</a>
1004	<a href="#">50 years, 50 colors</a>
1005	<a href="#">Card Game Cheater</a>
1006	<a href="#">Uncle Tom's Inherited Land*</a>
1007	<a href="#">Girls and Boys</a>
1008	<a href="#">Fire Net</a>
1009	<a href="#">Courses</a>
1010	<a href="#">过山车</a>

## 1001 Machine Schedule

### AC代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[510][510];/*连接关系*/
4  int vis[510];/*右边点使用与否*/
5  int link[510];/*右边点连接的左边点*/
6  int n, m, k;
7  int dfs(int x)
8  {
9      for (int y = 0; y < m; y++) {
10         if (!vis[y] && con[x][y]) {
11             vis[y] = 1;
12             if (dfs(link[y]) || link[y] == -1) {
13                 link[y] = x;
14                 return 1;
15             }
16         }
17     }
18     return 0;
19 }
20 int hungary()
21 {
22     int res = 0;
```

```

23     memset(link, -1, sizeof(link));
24     for (int x = 0; x < n; x++) {
25         memset(vis, 0, sizeof(vis));
26         if (dfs(x))res++;
27     }
28     return res;
29 }
30 int main()
31 {
32     while (scanf("%d", &n), n) {
33         scanf("%d%d", &m, &k);
34         int id, x, y;
35         memset(con, 0, sizeof(con));
36         for (int i = 0; i < k; i++) {
37             scanf("%d%d%d", &id, &x, &y);
38             if (x && y)con[x][y] = 1;
39         }
40         printf("%d\n", hungary());
41     }
42     return 0;
43 }

```

## 1002 Air Raid

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[150][150];/*连接关系*/
4  int vis[150];/*右边点使用与否*/
5  int link[150];/*右边点连接的左边点*/
6  int n, m, k;/*如题意*/
7  int dfs(int x)/*用以判断找不找得到结尾为之前未连的边的情况*/
8  {
9      for (int y = 1; y <= m; y++) {
10         if (!vis[y] && con[x][y]) /*代表右边这个点在此轮没有使用过且可有连线*/
11         {
12             vis[y] = 1;/*标注已使用*/
13             if (dfs(link[y]) || link[y] == -1) /*代表要么这个点并未连过或者它的连
14             线的点能找到另一个有未连的边的点.....*/
15             {
16                 link[y] = x;/*找到就重新连接*/
17                 return 1;
18             }
19         }
20     }
21     return 0;
22 }
23 int hungary()
24 {
25     int res = 0;
26     memset(link, -1, sizeof(link));/*注意连线是需要存储的，不然下一步就没有方向了*/
27     for (int x = 1; x <=n; x++) {
28         memset(vis, 0, sizeof(vis));/*注意这是每轮清空，表示需要抢之前已经匹配的*/

```

```

28         if (dfs(x))res++;/*成功一次多一个匹配数*/
29     }
30     return n-res;
31 }
32 int t;
33 int main()
34 {
35     scanf("%d", &t);
36     while (t--) {
37         scanf("%d%d", &n, &k);
38         m = n;
39         int x, y;
40         memset(con, 0, sizeof(con));
41         for (int i = 1; i <=k; i++) {
42             scanf("%d%d", &x, &y);
43             con[x][y] = 1;
44         }
45         printf("%d\n", hungary());
46     }
47     return 0;
48 }

```

## 1003 棋盘游戏

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[110][110];
4  int vis[110];
5  int link[110];
6  int n, m;
7  int find(int x)
8  {
9      for (int i = 1; i <= m; i++)
10     {
11         if (vis[i] == 0 && con[x][i] == 1)
12         {
13             vis[i] = 1;
14             if (link[i] == -1 || find(link[i]))
15             {
16                 link[i] = x;
17                 return 1;
18             }
19         }
20     }
21     return 0;
22 }
23 int hungary()
24 {
25     int res = 0;
26     memset(link, -1, sizeof(link));
27     for (int x = 1; x <= n; x++) {
28         memset(vis, 0, sizeof(vis));

```

```

29         if (find(x))res++;
30     }
31     return res;
32 }
33 int x[10010], y[10010];/*注意这里数组大小是k的范围*/
34 int main()
35 {
36     int k;
37     int count = 1;
38     while (scanf("%d%d%d", &n, &m, &k) == 3) {
39         memset(con, 0, sizeof(con));
40         for (int i = 1; i <= k; i++) {
41             scanf("%d%d", &x[i], &y[i]);
42             con[x[i]][y[i]] = 1;
43         }
44         int sum = hungary();
45         int cnt = 0;
46         for (int i = 1; i <= k; i++) {
47             con[x[i]][y[i]] = 0;
48             if (hungary() < sum)cnt++;
49             con[x[i]][y[i]] = 1;
50         }
51         printf("Board %d have %d important blanks for %d chessmen.\n",
count++, cnt, sum);
52     }
53     return 0;
54 }

```

## 1004 50 years,50 colors

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[110][110];
4  int vis[110];
5  int link[110];
6  int n;
7  int find(int x)
8  {
9      for (int i = 1; i <= n; i++)
10     {
11         if (vis[i] == 0 && con[x][i] == 1)
12         {
13             vis[i] = 1;
14             if (link[i] == -1 || find(link[i]))
15             {
16                 link[i] = x;
17                 return 1;
18             }
19         }
20     }
21     return 0;
22 }

```

```

23 int hungary()
24 {
25     int res = 0;
26     memset(link, -1, sizeof(link));
27     for (int x = 1; x <= n; x++) {
28         memset(vis, 0, sizeof(vis));
29         if (find(x))res++;
30     }
31     return res;
32 }
33 int Map[110][110];
34 int visc[55];
35 int main()
36 {
37     int k;
38     int count = 1;
39     while (scanf("%d%d", &n,&k),n,k) {
40         memset(con, 0, sizeof(con));
41         memset(visc, 0, sizeof(visc));
42         for (int i = 1; i <= n; i++) {
43             for (int j = 1; j <= n; j++) {
44                 scanf("%d", &Map[i][j]);
45                 visc[Map[i][j]] = 1;
46             }
47         }
48         int flag = 1;
49         for (int q = 1; q <= 50; q++) {
50             if (!visc[q])continue;
51             memset(con, 0, sizeof(con));
52             for(int i=1;i<=n;i++)
53                 for (int j = 1; j <= n; j++) {
54                     if (Map[i][j] == q)con[i][j] = 1;
55                 }
56             if (hungary() > k) {
57                 if (flag)printf("%d", q);
58                 else printf(" %d", q);
59                 flag = 0;
60             }
61         }
62         if (flag)printf("-1");
63         printf("\n");
64     }
65     return 0;
66 }

```

## 1005 Card Game Cheater

### AC代码

```

1 #include<bits/stdc++.h>
2 using namespace std;
3 int con[110][110];
4 int vis[110];
5 int link[110];

```

```

6  int n;
7  int t;
8  int find(int x)
9  {
10     for (int i = 1; i <= n; i++){
11         if (vis[i] == 0 && con[x][i] == 1){
12             vis[i] = 1;
13             if (link[i] == -1 || find(link[i])){
14                 link[i] = x;
15                 return 1;
16             }
17         }
18     }
19     return 0;
20 }
21 int hungary()
22 {
23     int res = 0;
24     memset(link, -1, sizeof(link));
25     for (int x = 1; x <= n; x++) {
26         memset(vis, 0, sizeof(vis));
27         if (find(x))res++;
28     }
29     return res;
30 }
31
32 int getScore(string str) {
33     int ans = 0;
34     if (str[0] >= '2' && str[0] <= '9')
35         ans += (str[0] - '0') * 10;
36     else if (str[0] == 'T')
37         ans += 100;
38     else if (str[0] == 'J')
39         ans += 110;
40     else if (str[0] == 'Q')
41         ans += 120;
42     else if (str[0] == 'K')
43         ans += 130;
44     else if (str[0] == 'A')
45         ans += 140;
46     if (str[1] == 'C')
47         ans += 1;
48     else if (str[1] == 'D')
49         ans += 2;
50     else if (str[1] == 'S')
51         ans += 3;
52     else if (str[1] == 'H')
53         ans += 4;
54     return ans;
55 }
56
57 int main()
58 {
59     scanf("%d", &t);
60     while (t--) {
61         memset(con, 0, sizeof(con));
62         scanf("%d", &n);
63         string str;

```

```

64     int scoreAdam[2000], scoreEve[2000];
65     for (int i = 1; i <= n; i++) {
66         cin >> str;
67         scoreAdam[i] = getScore(str);
68     }
69     for (int i = 1; i <= n; i++) {
70         cin >> str;
71         scoreEve[i] = getScore(str);
72     }
73     for (int i = 1; i <= n; i++)
74         for (int j = 1; j <= n; j++)
75             if (scoreEve[i] > scoreAdam[j])
76                 con[i][j] = 1;
77     printf("%d\n", hungary());
78 }
79 return 0;
80 }

```

---

## 1006 Uncle Tom's Inherited Land\*

未AC

---

## 1007 Girls and Boys

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[510][510];
4  int vis[510];
5  int link[510];
6  int n, m;
7  int find(int x)
8  {
9      for (int i = 0; i < n; i++)
10     {
11         if (vis[i] == 0 && con[x][i] == 1)
12         {
13             vis[i] = 1;
14             if (link[i] == -1 || find(link[i]))
15             {
16                 link[i] = x;
17                 return 1;
18             }
19         }
20     }
21     return 0;
22 }
23
24 int main()
25 {
26     while (scanf("%d", &n) != EOF) {

```

```

27     m = n;
28     memset(con, 0, sizeof(con));
29     memset(link, -1, sizeof(link));
30     int x,p,y;
31     for (int i = 0; i < n; i++) {
32         scanf("%d: (%d)", &x, &p);
33         while (p--) {
34             scanf("%d", &y);
35             con[x][y] = 1;
36         }
37     }
38     int res = 0;
39     for (int x = 0; x < n; x++) {
40         memset(vis, 0, sizeof(vis));
41         if (find(x))res++;
42     }
43     printf("%d\n", n - res / 2);
44 }
45 return 0;
46 }

```

## 1008 Fire Net

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[10][10];
4  int vis[10];
5  int link[10];
6  int n;
7  char Map[10][10];
8  int find(int x)
9  {
10     for (int i = 0; i < n*2; i++){
11         if (vis[i] == 0 && con[x][i] == 1){
12             vis[i] = 1;
13             if (link[i] == -1 || find(link[i])){
14                 link[i] = x;
15                 return 1;
16             }
17         }
18     }
19     return 0;
20 }
21 int hungary()
22 {
23     int res = 0;
24     memset(link, -1, sizeof(link));
25     for (int x = 0; x < n*2; x++) {
26         memset(vis, 0, sizeof(vis));
27         if (find(x))res++;
28     }
29     return res;

```



```

30 }
31
32
33 int main()
34 {
35     while (~scanf("%d", &n) && n) {
36         memset(con, 0, sizeof(con));
37         for (int i = 0; i < n; ++i) {
38             scanf("%s", Map[i]);
39         }
40         for (int i = 0; i < n; ++i) {
41             int flag = 0;
42             int flag1 = 0;
43             for (int j = 0; j < n; ++j) {
44                 if (Map[i][j] == '.' && flag == 1) {
45                     flag = 0;
46                     con[n + i][j] = 1;
47                 }
48                 else if (Map[i][j] == '.' && flag == 0) {
49                     con[i][j] = 1;
50                 }
51                 else if (Map[i][j] == 'x') {
52                     flag = 1;
53                 }
54                 if (Map[j][i] == '.' && flag1 == 1) {
55                     flag1 = 0;
56                     con[j][i + n] = 1;
57                 }
58                 else if (Map[j][i] == '.' && flag1 == 0) {
59                     con[j][i] = 1;
60                 }
61                 else if (Map[j][i] == 'x') {
62                     flag1 = 1;
63                 }
64             }
65         }
66         printf("%d\n", hungary());
67     }
68     return 0;
69 }

```

## 1009 Courses

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[305][305];
4  int vis[305];
5  int link[305];
6  int n, m;
7
8  int find(int x)
9  {

```

```

10     for (int i = 1; i <= m; i++){
11         if (vis[i] == 0 && con[x][i] == 1){
12             vis[i] = 1;
13             if (link[i] == -1 || find(link[i])){
14                 link[i] = x;
15                 return 1;
16             }
17         }
18     }
19     return 0;
20 }
21 int hungary()
22 {
23     int res = 0;
24     memset(link, -1, sizeof(link));
25     for (int x = 1; x <= n; x++) {
26         memset(vis, 0, sizeof(vis));
27         if (find(x))res++;
28     }
29     return res;
30 }
31
32 int t;
33 int k;
34 int main()
35 {
36     scanf("%d", &t);
37     while (t--) {
38         memset(con, 0, sizeof(con));
39         scanf("%d%d", &n, &m);
40         int g;
41         for (int i = 1; i <= n; i++) {
42             scanf("%d", &g);
43             int y;
44             while (g--) {
45                 scanf("%d", &y);
46                 con[i][y] = 1;
47             }
48         }
49         if (hungary() == n)printf("YES\n");
50         else printf("NO\n");
51     }
52     return 0;
53 }

```

## 1010 过山车

### AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int con[510][510];
4  int vis[510];
5  int link[510];

```

```
6  int n, m, k;
7  int dfs(int x)
8  {
9      for (int y = 1; y <= m; y++) {
10         if (!vis[y] && con[x][y])
11             {
12                 vis[y] = 1;
13                 if (dfs(link[y]) || link[y] == -1)
14                     {
15                         link[y] = x;
16                         return 1;
17                     }
18             }
19     }
20     return 0;
21 }
22 int hungary()
23 {
24     int res = 0;
25     memset(link, -1, sizeof(link));
26     for (int x = 1; x <= n; x++) {
27         memset(vis, 0, sizeof(vis));
28         if (dfs(x)) res++;
29     }
30     return res;
31 }
32 //int t;
33 int main()
34 {
35     while (scanf("%d",&k),k) {
36         scanf("%d%d", &n, &m);
37         int x, y;
38         memset(con, 0, sizeof(con));
39         for (int i = 1; i <= k; i++) {
40             scanf("%d%d", &x, &y);
41             con[x][y] = 1;
42         }
43         printf("%d\n", hungary());
44     }
45     return 0;
46 }
```

---