

专题3

目录

Problem ID	Title
1000	Sticks
1001	N皇后问题
1002	Sudoku Killer
1003	Card Game Cheater
1004	连连看
1005	Tempter of the Bone
1006	Red and Black
1007	排列2
1008	Sudoku
1009	变形课
1010	Friends
1011	How many ways
1012	FatMouse and Cheese

1000 Sticks

AC代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int cmp(int x, int y)
4  {
5      return x > y; /*改变sort方向*/
6  }
7  int a[70];
8  int vis[70]; /*存放是否使用过的状态*/
9  int n; /*总量*/
10 int ans; /*各组长度和*/
11 int dfs(int cnt, int cur, int pos);
12 /*已选择数；当前组已选择长度和；当前位置*/
13 int main()
14 {
15     while (scanf("%d", &n), n) {
16         int sum = 0;
```

```

17     for (int i = 0; i < n; i++) {
18         scanf("%d", &a[i]);
19         sum += a[i];
20     }
21     sort(a, a + n, cmp);
22     for (int i = n; i >= 1; i--) {
23         if (sum % i == 0 && sum / i >= a[0]) {
24             ans = sum / i; /*剪枝*/
25             memset(vis, 0, sizeof(vis));
26             if (dfs(0, 0, 0)) {
27                 printf("%d\n", ans);
28                 break;
29             }
30         }
31     }
32     return 0;
33 }
34
35 int dfs(int cnt, int cur, int pos)
36 {
37     if (cnt == n) return 1;
38     for (int i = pos; i < n; i++) {
39         if (vis[i]) continue;
40         if (cur + a[i] < ans) {
41             vis[i] = 1;
42             if (dfs(cnt + 1, cur + a[i], i + 1)) return 1; /*不要将i+1误写成
pos+1*/
43             vis[i] = 0;
44             if (cur == 0) return 0; /*剪枝：往后棍子更少更凑不起来*/
45             while (a[i + 1] == a[i] && i + 1 < n) i++; /*剪枝：试过的不再试*/
46         }
47         else if (cur + a[i] == ans) {
48             vis[i] = 1;
49             if (dfs(cnt+1, 0, 0)) return 1;
50             vis[i] = 0; /*不要漏写了回溯*/
51             return 0;
52         }
53     }
54     return 0;
55 }

```

1001 N皇后问题

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #include<bits/stdc++.h>
4  using namespace std;
5  int a[15]; /*列*/
6  int c[30]; /*副对角线*/
7  int d[30]; /*主对角线*/
8  int num[15];
9  int n;

```

```

10  int cnt;
11  int text[11];
12  void dfs(int i);
13  int main()
14  {
15      for (n = 1; n <= 10; n++) {
16          cnt = 0;
17          dfs(1);
18          text[n] = cnt; /*需要打表储存一下，不然会超时*/
19      }
20      while (scanf("%d", &n), n) {
21          printf("%d\n", text[n]);
22      }
23      return 0;
24  }
25  void dfs(int i)
26  {
27      if (i == n + 1) {
28          cnt++;
29          return;
30      }
31      for (int j = 1; j <= n; j++)
32      {
33          if (a[j] == 0 && c[i + j - 1] == 0 && d[n - i + j] == 0) {
34              num[i] = j;
35              a[j] = 1; c[i + j - 1] = 1; d[n - i + j] = 1;
36              dfs(i + 1);
37              a[j] = 0; c[i + j - 1] = 0; d[n - i + j] = 0;
38          }
39      }
40      return;
41  }

```

1002 Sudoku Killer

未AC

1003 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 }

```

1004 连连看

AC代码

1005 Tempter of the Bone

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int n, m, t;//n行m列时间t
4  char Map[10][10];
5  int se[4][2] = { {-1,0},{1,0},{0,-1},{0,1} };//移动方向
6  int flag;
7  int dx, dy;//结束点*/
8  int sx, sy;//初始点*/
9
10 void dfs(int sx,int sy,int time)
11 {
12     int temp;
13     if (sx > n || sy > m || sx < 1 || sy < 1)return;/*注意sx, sy的范围，千万不能
14     写错*/
15     if (time == t && dx == sx && dy == sy) flag = 1;
16     if (flag)return;
17     temp = (t - time) - abs(sx - dx) - abs(sy - dy);
18     if (temp < 0 || temp % 2 == 1) return;/*奇偶性剪枝，不然此题超时*/
19     for (int i = 0; i < 4; i++)
20     {
21         if (Map[sx + se[i][0]][sy + se[i][1]] != 'X')
22         {
23             Map[sx + se[i][0]][sy + se[i][1]] = 'X';
24             dfs(sx + se[i][0], sy + se[i][1], time + 1);
25             Map[sx + se[i][0]][sy + se[i][1]] = '.';/*回溯*/
26         }
27     }
28     return;
29 }
30 int main()
31 {
32     while (scanf("%d%d%d", &n, &m, &t))

```

```

32     {
33         flag = 0;
34         int wall = 0;
35         if (m == 0 && n == 0 && t == 0) break;
36         for (int i = 1; i <= n; i++)
37             for (int j = 1; j <= m; j++)
38             {
39                 cin >> Map[i][j];
40                 if (Map[i][j] == 'S') {
41                     sx = i; sy = j;
42                 }
43                 if (Map[i][j] == 'D') {
44                     dx = i; dy = j;
45                 }
46                 if (Map[i][j] == 'X')
47                     wall++;
48             }
49         if (m * n - wall <= t)
50             printf("NO\n");
51         else {
52             Map[sx][sy] = 'X';
53             dfs(sx, sy, 0);
54             if (flag) printf("YES\n");
55             else printf("NO\n");
56         }
57     }
58     return 0;
59 }

```

1006 Red and Black

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int w, h;
4  char Map[22][22];
5  int vis[22][22];
6  int dis[4][2] = { -1,0,1,0,0,1,0,-1 };
7  int cnt;
8  int st, ed;
9  void dfs(int x, int y);
10 int main()
11 {
12     while (scanf("%d%d", &w, &h), w, h) {
13         memset(vis, 0, sizeof(vis));
14         for (int i = 1; i <= h; i++)
15             for (int j = 1; j <= w; j++)
16             {
17                 cin >> Map[i][j];
18                 if (Map[i][j] == '@') {
19                     st = i, ed = j;
20                 }
21             }

```

```

22     cnt = 0;
23     dfs(st, ed);
24     printf("%d\n", cnt);
25 }
26 return 0;
27 }
28 void dfs(int x, int y)
29 {
30     cnt++;
31     vis[x][y] = 1;
32     for (int i = 0; i < 4; i++) {
33         int xx = x + dis[i][0];
34         int yy = y + dis[i][1];
35         if (!vis[xx][yy] && xx >= 1 && xx <= h && yy >= 1 && yy <= w) {
36             if (Map[xx][yy] != '#')
37                 dfs(xx, yy);
38         }
39     }
40 }

```

1007 排列2

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int a[4];
4  int main()
5  {
6      scanf("%d%d%d%d", &a[0], &a[1], &a[2], &a[3]);
7      while (1) {
8          if (a[0] + a[1] + a[2] + a[3] == 0)break;
9          int temp = a[0];
10         if (a[0]) {
11             for (int i = 0; i < 4; i++)printf("%d", a[i]);
12         }
13         while (next_permutation(a, a + 4)) {/*对数组中对应地址的数排列*/
14             if (a[0]) {
15                 if (a[0] != temp) {
16                     if (temp)printf("\n");
17                     temp = a[0];
18                 }
19                 else printf(" ");
20                 for (int i = 0; i < 4; i++)
21                     printf("%d", a[i]);
22             }
23         }
24         printf("\n");/*注意题目输出格式*/
25         scanf("%d%d%d%d", &a[0], &a[1], &a[2], &a[3]);
26         if (a[0] + a[1] + a[2] + a[3])printf("\n");/*注意题目输出格式*/
27     }
28     return 0;
29 }

```

1008 Sudoku

未AC

1009 变形课

AC代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  struct turn {
4      char fr; /*首字母*/
5      char to; /*尾字母*/
6      int flag; /*是否使用过*/
7  }s[10000];
8  char a[10000];
9  int cnt; /*记录法则数量*/
10 void bfs();
11 int main()
12 {
13     while (scanf("%s", a) != EOF) {
14         int len = strlen(a);
15         cnt = 0;
16         s[++cnt].fr = a[0];
17         s[cnt].to = a[len - 1];
18         s[cnt].flag = 0;
19         while (++cnt) {
20             scanf("%s", a);
21             if (a[0] == '0') break;
22             len = strlen(a);
23             s[cnt].fr = a[0];
24             s[cnt].to = a[len - 1];
25             s[cnt].flag = 0;
26         }
27         bfs();
28     }
29     return 0;
30 }
31 void bfs()
32 {
33     queue<turn>qu;
34     turn cur, nex;
35     cur.to = 'b';
36     cur.flag = 1;
37     qu.push(cur);
38     while (!qu.empty()) {
39         cur = qu.front();
40         qu.pop();
41         if (cur.to == 'm') {
42             printf("Yes.\n");
43             return;
44         }
45         for (int i = 1; i <= cnt - 1; i++) {
```



```

46         if (s[i].fr == cur.to && s[i].flag == 0) {
47             nex = s[i];
48             s[i].flag = 1;
49             qu.push(nex);
50         }
51     }
52 }
53 printf("No.\n");
54 }

```

1010 Friends

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int t;
4  int n, m;
5  int con[30][2];/*朋友对应关系*/
6  int on[10], off[10];/*每个人线上线下朋友数*/
7  int ti[10];
8  int cnt;
9  /*测试: int test = 0;*/
10 void dfs(int k);
11 int check(int s1[], int s2[],int p)/*判断是否朋友数相等*/
12 {
13     for (int i = 1; i <= p; i++)
14         if (s1[i] != s2[i])return 0;
15     return 1;
16 }
17 int main()
18 {
19     scanf("%d", &t);
20     while (t--) {
21         cnt = 0;
22         memset(on, 0, sizeof(on));
23         memset(off, 0, sizeof(off));
24         memset(ti, 0, sizeof(ti));
25         scanf("%d%d", &n, &m);
26         for (int i = 1; i <= m; i++) {
27             scanf("%d%d", &con[i][0], &con[i][1]);
28             ti[con[i][0]]++;
29             ti[con[i][1]]++;
30         }
31         int flag = 1;
32         for (int i = 1; i <= n; i++) {
33             if (ti[i] % 2 == 1) {
34                 printf("0\n");
35                 flag = 0;
36                 break;
37             }
38         }
39         if (flag) {
40             dfs(1);

```

```

41         printf("%d\n", cnt);
42     }
43 }
44 return 0;
45 }
46 void dfs(int k)/*k为已用关系数+1*/
47 {
48     if (k == m + 1) {
49         /*测试程序*/
50         /*printf("%d\n", ++test);
51         for (int i = 1; i <= n; i++) {
52             printf("on:%d,off:%d\n", on[i], off[i]);
53         }
54         printf("%d\n", check(on, off, n));
55         printf("\n\n");*/
56         if (check(on, off, n))cnt++;
57         return;
58     }
59     /*剪枝*/
60     if (on[con[k][0]] < ti[con[k][0]] / 2 && on[con[k][1]] < ti[con[k][1]] /
2) {
61         on[con[k][0]]++;/*对应朋友数加1*/
62         on[con[k][1]]++;
63         dfs(k + 1);
64         on[con[k][0]]--;/*如果on不行，退回上一步选off*/
65         on[con[k][1]]--;
66     }
67     if (off[con[k][0]] < ti[con[k][0]] / 2 && off[con[k][1]] < ti[con[k][1]] /
/ 2) {
68         off[con[k][0]]++;
69         off[con[k][1]]++;
70         dfs(k + 1);
71         off[con[k][0]]--;
72         off[con[k][1]]--;
73     }
74     return;
75 }

```

1011 How many ways

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int Map[105][105];
4  int Sum[105][105];
5  int t, m, n;
6  int si, sj;
7  int dfs(int si, int sj);
8  int main()
9  {
10     scanf("%d", &t);
11     while (t--)
12     {

```

```

13     memset(Sum,0,sizeof(Sum));
14     scanf("%d%d", &n, &m);
15     for(int i=1;i<=n;i++)
16         for (int j=1; j <= m; j++)
17             scanf("%d", &Map[i][j]);
18     si = 1; sj = 1;
19     printf("%d\n", dfs(1, 1)%10000);
20 }
21 return 0;
22 }
23 int dfs(int si, int sj)
24 {
25     if (Sum[si][sj])return Sum[si][sj];
26     int step = Map[si][sj];
27     if (si == n && sj == m) {
28         Sum[si][sj]=1; return 1;
29     }
30     int i, j;
31     for (i = 0; i <= step; i++)
32         for (j = 0; i + j <= step; j++)
33         {
34             if (i + j == 0)continue;
35             if (si+i<=n&&sj+j<=m) {
36                 Sum[si][sj]=(Sum[si][sj]+dfs(si + i, sj + j))%10000; /*将该点
后续情况数存放在数组里*/
37             }
38         }
39     return Sum[si][sj];
40 }

```

1012 FatMouse and Cheese

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int n, k;
4  int init[102][102]; /*初始值*/
5  int ans[102][102]; /*dfs储存值*/
6  int dis[4][2] = { 1,0,-1,0,0,1,0,-1 }; /*方向*/
7  int OK(int x, int y)
8  {
9      if (x < 0 || x >= n || y < 0 || y >= n)return 0; /*判断是否在区域边界内*/
10     else return 1;
11 }
12 int dfs(int x, int y);
13 int main()
14 {
15     while (scanf("%d%d", &n, &k), n != -1, k != -1) {
16         memset(ans, 0, sizeof(ans)); /*注意对ans数组的清零*/
17         for (int i = 0; i < n; i++)
18             for (int j = 0; j < n; j++)
19                 scanf("%d", &init[i][j]);
20         printf("%d\n", dfs(0,0));

```

```
21     }
22     return 0;
23 }
24 int dfs(int x, int y)
25 {
26     int answer = 0;
27     if (ans[x][y]) return ans[x][y]; /*代表该位置在其它路线上已经计算过*/
28     for(int i=0;i<4;i++)
29         for (int j = 1; j <= k; j++) {
30             int xx = x + dis[i][0] * j;
31             int yy = y + dis[i][1] * j;
32             if (OK(xx, yy) && init[xx][yy] > init[x][y])
33                 answer = max(answer, dfs(xx,yy)); /*由于题目要求求得最大值，保存最
大的情况*/
34         }
35     ans[x][y] = answer + init[x][y]; /*记忆化dfs*/
36     return ans[x][y];
37 }
```
