

原创

状压DP入门例题

2019-08-25 00:58:18    -Y- -Y-    阅读数 54    更多

编辑

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：[https://blog.csdn.net/weixin\\_44410512/article/details/100058465](https://blog.csdn.net/weixin_44410512/article/details/100058465)

互不侵犯King

```
1  /*
2   互不侵犯King
3   HYSBZ - 1087
4   https://cn.vjudge.net/problem/HYSBZ-1087
5   题意：n×n的矩阵里放k个国王，每个国王的周边8个格子不能放其他国王，问有多少方法
6   解法：
7   f[i][j][L] 表示前i行状态为j上有k个国王时的最优解
8   状压dp，用数字的二进制来表示这一行上的国王分布
9   然后记录在一行中，可以有效表达国王分布的数字并记录
10  设当前行的状态为j，上一行的状态为x，可以得到下面的转移方程：
11  f[i][j][L]=f[i-1][x][L-sta[x]]
12  */
13  #include <algorithm>
14  #include <iostream>
15  using namespace std;
16  #define maxn 2005
17  long long sit[maxn]; // 记录一行中所有的合法状态
18  long long sta[maxn]; // 记录合法状态下的国王数
19  long long f[15][maxn][105]; // f[i][j][L] 表示前i行状态为j上有k个国王时的最优解
20  int n, k, cnt;
21  void dfs(int x,int num,int cur){ // x表示这一行的状态 num记录国王数 cur记录状态长度
22      if(cur>=n){
23          sit[++cnt]=x;
24          sta[cnt]=num;
25          return ;
26      }
27      dfs(x,num,cur+1); // cur位置不放国王
28      dfs(x+(1<<cur),num+1,cur+2); // cur位置放国王，与它相邻的位置不能再放国王
29  }
30  int main(){
31      cin>>n>>k;
32      dfs(0,0,0); // 先预处理一行的所有合法状态
33      for(int i=1; i<=cnt;i++) f[1][i][sta[i]]++;
34      for(int i=2;i<=n;i++){
35          for(int j=1;j<=cnt;j++){
36              for(int l=1;l<=cnt;l++){
37                  if(sit[j]&sit[l]) continue;
38                  if((sit[j]<<1)&sit[l]) continue;
39                  if(sit[j]&(sit[l]<<1)) continue;
40                  // 排除不合法转移
41                  for(int p=sta[j];p<=k;p++){
42                      f[i][j][p]+=f[i-1][l][p-sta[j]];
43                  }
44              }
45          }
46      }
47      long long ans=0;
48      for(int i=1;i<=cnt;i++) ans+=f[n][i][k];
49      cout<<ans<<endl;
50      return 0;
51  }
```

炮兵阵地

```
1  /*
2   炮兵阵地
3   P2704
```

编程语言大PK，你选谁？

关闭

```

4  https://www.luogu.org/problem/P2704
5  解法:
6  状压DP
7  由于当前行和前两行有关系, 所以得用3维矩阵来保存一个状态下最多的炮兵个数
8  dp[i][curst][prest] 表示当前第i行状态对curst, 前一行状态为prest的最大炮兵数
9  dp[i][curst][prest]=max{dp[i-1][prest][preprest]}
10  这样求到最后一行之后, 答案就是最后一行所有状态中最大的那个。
11  程序初始化的时候需要对第一行
12  */
13  #include <bits/stdc++.h>
14  using namespace std;
15  int dp[115][70][70];
16  int maze[105];
17  int sit[70];
18  int sta[70];
19  int n,m,len;
20  char str[111];
21  int main(){
22      while(cin>>n>>m){
23          memset(dp,0,sizeof dp);
24          memset(maze,0,sizeof maze);
25          memset(sit,0,sizeof sit);
26          memset(sta,0,sizeof sta);
27          for(int i=1;i<=n;i++){
28              scanf("%s", str);
29              for(int j=0;j<m;j++){
30                  if(str[j]=='H'){
31                      maze[i]+=(1<<j);
32                  }
33              }
34          }
35          len=0;
36          for(int i=0;i<(1<<m);i++){
37              if(!(i&(i<<1)) && !(i&(i<<2))){
38                  int k=i,sum=0;
39                  while(k){
40                      if(k&1) sum++;
41                      k>>=1;
42                  }
43                  sta[++len]=sum;
44                  sit[len]=i;
45              }
46          }
47          for(int i=1;i<=len;i++){
48              if(!(sit[i]&maze[1])){
49                  dp[1][i][1]=sta[i];
50              }
51          }
52          for(int i=2;i<=n;i++){
53              for(int j=1;j<=len;j++){
54                  if(sit[j]&maze[i]) continue;
55                  for(int k=1;k<=len;k++){
56                      if(sit[k]&sit[j]) continue;
57                      if(sit[k]&maze[i-1]) continue;
58                      for(int t=1;t<=len;t++){
59                          if(sit[t]&sit[j]) continue;
60                          if(sit[t]&sit[k]) continue;
61                          if(sit[t]&maze[i-2]) continue;
62                          dp[i][j][k]=max(dp[i][j][k],dp[i-1][k][t]+sta[j]);
63                      }
64                  }
65              }
66          }
67          int ans=0;
68          for(int i=1;i<=len;i++){
69              for(int j=1;j<=len;j++){
70                  ans=max(ans,dp[n][i][j]);
71              }
72          }
73          cout<<ans<<endl;
74      }

```

```
75 | return 0;
76 | }
77 |
```

## 玉米田Corn Fields

```
1  /*
2   玉米田Corn Fields
3   P1879
4   https://www.luogu.org/problem/P1879
5   题面:
6   有一个N×M的田地, 0表示贫瘠, 1表示肥沃,
7   只能在肥沃的土地上种玉米,
8   并且每一颗玉米都不能相邻, 问有多少种方法
9   解法:
10  状压DP
11  */
12  #include <bits/stdc++.h>
13  using namespace std;
14  #define maxn 10000
15  #define MOD 100000000
16  long long dp[15][maxn];
17  long long sit[maxn];
18  long long maze[maxn];
19  int n,m,a,len;
20  int main(){
21      cin>>n>>m;
22      for(int i=1;i<=n;i++){
23          for(int j=0;j<m;j++){
24              cin>>a;
25              a=1-a;
26              maze[i]+=(a<<j);
27          }
28      }
29      len=0;
30      for(int i=0;i<(1<<m);i++){
31          if(!(i&(i<<1))){
32              sit[++len]=i;
33          }
34      }
35      dp[0][1]=1;
36      for(int i=1;i<=n;i++){
37          for(int j=1;j<=len;j++){
38              if(sit[j]&maze[i]) continue;
39              for(int k=1;k<=len;k++){
40                  if(sit[k]&sit[j]) continue;
41                  if(sit[k]&maze[i-1]) continue;
42                  dp[i][j]+=dp[i-1][k];
43                  dp[i][j]%=MOD;
44              }
45          }
46      }
47      long long ans=0;
48      for(int i=1;i<=len;i++){
49          ans+=dp[n][i];
50          ans%=MOD;
51      }
52      cout<<ans<<endl;
53      return 0;
54  }
```

有 0 个人打赏

文章最后发布于: 2019-08-25 00:58:18

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客

编程语言大PK，你选谁？

关闭