

状压dp by iuiou

状压的描述：所谓状压，其实就是状态压缩，就是把一些多维的状态用一个简洁的二进制数来表示。比如这样一个问题<https://www.luogu.com.cn/problem/P4011>

题目中明确指出 m 把钥匙，没把钥匙可以开一个对应的门，在迷宫中行走的时候，自己身上拥有钥匙的数量和种类肯定会变化，这给不管是dp还是bfs都带来了麻烦，所以考虑，设置一个 m 位2进制字符串，第 i 位是否为1表示当前是否有第 i 把钥匙，这样在bfs过程中可以通过位运算来改变当前的状态。这样就处理起来方便很多。

上题代码

```
#include <bits/stdc++.h> // 状压bfs , 讲不通的状态压缩成2进制代码
using namespace std;
int ck[15][15][15][15]; // 存两个各自之间是否存在墙|门或者是啥都没有
int vis[15][15][9024]; // 判断是否是两个完全相同的状态, 只有所处位置一样, 而且所拿药匙相同才会相同
int dir[4][2] = {1, 0, -1, 0, 0, 1, 0, -1};
struct node {
    int x, y, stp, key; // key表示目前拿到的钥匙
};
queue<node> q;
int key[15][15];
int n, m, p, k;
int bfs()
{
    q.push((node){1, 1, 0, key[1][1]});
    vis[1][1][key[1][1]] = 1;
    while(!q.empty())
    {
        node now = q.front();
        q.pop();
        for(int i = 0; i < 4; i++)
        {
            int
x = now.x + dir[i][0], y = now.y + dir[i][1], ky = now.key, stp = now.stp + 1;
            if(x <= 0 || y <= 0 || x > n || y > m) continue;
            if((ck[now.x][now.y][x][y] & ky) != ck[now.x][now.y][x][y])
continue;
            ky |= key[x][y];
            if(vis[x][y][ky]) continue;
            vis[x][y][ky] = 1;
            if(x == n && y == m) return stp;
            q.push((node){x, y, stp, ky});
        }
    }
    return -1;
}
int main()
```

```
{
scanf ("%d%d%d", &n, &m, &p);
scanf ("%d", &k);
int x1, y1, x2, y2, g, s;
for(int i=1; i<=k; i++)
{
scanf ("%d%d%d%d%d", &x1, &y1, &x2, &y2, &g);
if(g==0) ck[x1][y1][x2][y2]=ck[x2][y2][x1][y1]=1<<(p);
else ck[x1][y1][x2][y2]=ck[x2][y2][x1][y1]=1<<(g-1);
}
scanf ("%d", &s);
for(int i=1; i<=s; i++)
{
scanf ("%d%d%d", &x1, &y1, &g);
key[x1][y1]=1<<(g-1);
}
int ans=bfs();
printf ("%d", ans);
}
```

而dp就是状态的转移，状压dp就是将状态做01压缩后成为dp数组的一维，然后做转移而已。具体怎么设定数组和怎么转移还要具体问题具体分析。

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:%E7%8A%B6%E5%8E%8Bdp&rev=1594887890>

Last update: 2020/07/16 16:24