



# C++ 算法

# 深度优先搜索

## Depth-first Search

棋盘格  
最短路问题

# 例题：最短路四方向有障碍版

在一个 $n*m$ 格子的迷宫里，o代表空地可以行走，#代表墙体不可以通过。从左上角(1,1)开始，每一步可以上下左右四个方向走动一格，请问需要至少几步可以达到(n,m)， $n,m \leq 10$

输入样例：

2 2

oo

#o

输出样例：

2

输入样例：

4 5

o#ooo

o#o#o

o#o#o

ooo#o

输出样例：

13

思考：假设只能  
向右和向下走呢

思考：如何用  
dfs 求解最短路

# 棋盘格最短路问题

算法1

全枚举所有路径，比较最优解

算法2

记录当前最优解，减少枚举量

# 算法1：全枚举

`dfs(x,y,len)`  
从(x,y)格子继续走  
已经走了len步

用dfs枚举所有可能的路径

根据每一条可行路径的长度，更新最短路径

```

1  #include<iostream>
2  #include<algorithm>
3  #define N 20
4  #define INF 10000
5  using namespace std;
6  int dx[4]={0,1,0,-1};
7  int dy[4]={1,0,-1,0};
8  int n,m,ans=INF;
9  char f[N][N];
10 bool vst[N][N];
11 void dfs(int x,int y,int len){
12     vst[x][y]=1;
13     if(x==n&&y==m){ans=min(ans,len);return;}
14     for(int k=0;k<4;k++){
15         int nx=x+dx[k],ny=y+dy[k];
16         if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&f[nx][ny]=='o'&&!vst[nx][ny])
17             dfs(nx,ny,len+1);
18     }
19     vst[x][y]=0;
20 }
21 int main(){
22     cin>>n>>m;
23     for(int i=1;i<=n;i++)
24         for(int j=1;j<=m;j++)
25             cin>>f[i][j];
26     dfs(1,1,0);

```

# 错误代码

```
6 int dx[4]={0,1,0,-1};
7 int dy[4]={1,0,-1,0};
8 int n,m,ans=INF;
9 char f[N][N];
10 void dfs(int x,int y,int len){
11     if(x==n&&y==m){ans=min(ans,len);return;}
12     for(int k=0;k<4;k++){
13         int nx=x+dx[k],ny=y+dy[k];
14         if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&f[nx][ny]=='o')
15             dfs(nx,ny,len+1);
16     }
17 }
```

程序WA输出错误时，  
若找不到代码问题，  
该如何定位错误？

# dfs调试方法

```
6 int dx[4]={0,1,0,-1};
7 int dy[4]={1,0,-1,0};
8 int n,m,ans=INF;
9 char f[N][N];
10 void dfs(int x,int y,int len){
11     cout<<x<<" "<<y<<" "<<len<<endl;
12     string s; getline(cin,s);
13     if(x==n&&y==m){ans=min(ans,len);return;}
14     for(int k=0;k<4;k++){
15         int nx=x+dx[k],ny=y+dy[k];
16         if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&f[nx][ny]=='o')
17             dfs(nx,ny,len+1);
18     }
19 }
```

打印日志  
输出关键变量



# 算法2： 记录当前最优解

`dfs(x,y,len)`  
从(x,y)格子继续走  
已经走了len步

全局数组`dist[x][y]`  
记录到(x,y)当前最少几步

调用`dfs(x,y,len)`时

如果`len >= dist[x][y]`， 直接返回

否则如果`len < dist[x][y]`， 更新`dist[x][y]`为len

```
1  #include<iostream>
2  #include<algorithm>
3  #define N 20
4  #define INF 10000
5  using namespace std;
6  int dx[4]={0,1,0,-1};
7  int dy[4]={1,0,-1,0};
8  int n,m,dist[N][N];
9  char f[N][N];
10 void dfs(int x,int y,int len){
11     if(len>=dist[x][y])return;
12     dist[x][y]=len;
13     if(x==n&&y==m)return;
14     for(int k=0;k<4;k++){
15         int nx=x+dx[k],ny=y+dy[k];
16         if(nx>=1&&nx<=n&&ny>=1&&ny<=m&&f[nx][ny]=='o')
17             dfs(nx,ny,len+1);
18     }
19 }
20 int main(){
```

# 记忆化搜索

dfs搜索过程中拥有“记忆”功能

dfs(x,y,len)参数中的len记录当前路径已经走了几步

全局数组vst[x][y]记录当前路径是否已经访问(x,y)格子

全局数组dist[x][y]记录已知路径访问(x,y)格子最少几步

# 例题：无人驾驶

在一个 $n*m$ 格子的迷宫里，`o`代表空地可以行走，`#`代表墙体，一辆无人驾驶汽车从左上角 $(1,1)$ 的空地出发，希望到达 $(n,m)$ 格子，每走一格可以上下左右四个方向前进。该车钢板挺厚的，有一次撞破墙的机会。在该车想走的格子里被墙挡住去路时，可以把一个格子里的墙撞毁，从而走入该格子。请问需要至少几步能到达目标？如果无法完成输出`mission impossible`

输入 $n$ 和 $m$ ， $n,m \leq 10$ 。接着是 $n*m$ 格字符。输出`mission impossible`或一个整数

输入样例：

3 5

o#ooo

o#o#o

ooo#o

输出样例：

6

输入样例：

4 5

o#ooo

o#o#o

ooo##

o###o

输出样例：

11

思考：如何用  
dfs 求解最短路

```

1 #include<iostream>
2 #include<algorithm>
3 #define N 20
4 #define INF 10000
5 using namespace std;
6 int dx[4]={0,1,0,-1};
7 int dy[4]={1,0,-1,0};
8 int n,m,dist[N][N];
9 char f[N][N];

```

尝试理解每一句代码

```

22 int main(){
23     cin>>n>>m;
24     for(int i=1;i<=n;i++)
25         for(int j=1;j<=m;j++)
26             cin>>f[i][j];
27     fill(dist[0],dist[0]+N*N,INF);
28     dfs(1,1,1,0);
29     int ans=dist[n][m];
30     if(ans<INF)cout<<ans;
31     else cout<<"mission impossible"<<endl;
32     return 0;
33 }

```

f[i][j]代表什么含义

dist[i][j]代表什么含义

dfs(1,1,1,0)四个参数  
代表什么含义

# 错误代码

```
10 void dfs(int x,int y,bool life,int len){
11     if(len>=dist[x][y])return;
12     dist[x][y]=len;
13     if(x==n&&y==m)return;
14     for(int k=0;k<4;k++){
15         int nx=x+dx[k],ny=y+dy[k];
16         if(nx>=1&&nx<=n&&ny>=1&&ny<=m){
17             if(f[nx][ny]=='o')dfs(nx,ny,life,len+1);
18             else if(life)dfs(nx,ny,0,len+1);
19         }
20     }
21 }
```

程序WA输出错误时，  
若找不到代码问题，  
该如何定位错误？