7.DFS

1、模板

```
int mp[100][100],vis[100][100];
bool flag = false;

int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

void dfs(int x, int y)  //传入参数是当前的坐标，注意起点位置是否合法
{
  if (vis[x][y])  return;  //表示已经访问过了 递归回溯
  vis[x][y] = true;

  if( x == n && y == n)  //递归终止
    return;

  for (int i = 0; i < 4; i++)
  {
      int nx = x + dx[i];
      int ny = y + dy[i];
      if (nx > 0 && nx <= H && ny > 0 && ny <= W && !vis[nx][ny] && mp[nx][r
        dfs(nx,ny);
  }

  vis[x][y] = false;
}
```

2、中国象棋

```
#include <iostream>
#include <stack>
using namespace std;
char s[10][10];
bool vis[10][10];
bool flag = false;

int dx[8] = {2,2,1,1,-1,-1,-2,-2};
int dy[8] = {1,-1,2,-2,2,-2,1,-1};

void dfs(int x, int y)
{
    if (vis[x][y])  return;
```

```
    vis[x][y] = true;
    if (s[x][y] == 'T')
    {
        flag =true;
        return;
    }
    for (int i = 0; i < 8; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <10 && ny >= 0 && ny < 10 && s[nx][ny] !='#' && !v
        {
            dfs(nx,ny);
        }
    }
}

int main()
{
    int x,y;
    for (int i = 0; i < 10; i++)
    {
        scanf("%s",s[i]);
    }
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (s[i][j] == 'S')
            {
                x = i;
                y = j;
            }
        }
    }
    dfs(x,y);
    if (flag)
        printf("YES");
    else
        printf("NO");

    return 0;
}
```

3、迷宫最短路

```
#include <iostream>
#include <string>
using namespace std;
int n,m;
string maze[110];
```

```
bool vis[110][110];
bool flag = false;
int ans = 1000000;
int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

void dfs(int x, int y,int step)
{
    if (vis[x][y])  return;
    vis[x][y] = true;
    if (maze[x][y] == 'T')
    {
      if (step < ans)
        ans = step;
      return;
    }
    for (int i = 0; i < 4; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <n && ny >= 0 && ny < m && maze[nx][ny] !='*' && !
        {
            dfs(nx,ny,step+1);
        }
    }
  vis[x][y] = false;
}

int main()
{
    int x,y;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        cin >> maze[i];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (maze[i][j] == 'S')
            {
                x = i;
                y = j;
            }
        }
    }
    dfs(x,y,0);
    cout << ans <<endl;

    return 0;
```

```
}
```

4、踏青

```
测试样例
5 6
.#....
..#...
..#..#
...##.
.#....
```

```
#include <iostream>
#include <string>
using namespace std;
int n,m;
string mp[105];
bool vis[105][105];
bool flag = false;
int cnt = 0;
int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

void dfs(int x, int y)
{
    if (vis[x][y])  return;
    vis[x][y] = true;
    for (int i = 0; i < 4; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <n && ny >= 0 && ny < m && mp[nx][ny] !='.' && !v
        {
            dfs(nx,ny);
        }
    }
}

int main()
{
    int x,y;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        cin >> mp[i];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
```

```
            if (mp[i][j] == '#' && !vis[i][j])
            {
                dfs(i,j);
                cnt ++;
            }
        }
    }
    cout << cnt <<endl;

    return 0;
}
```

5、迷宫解的方案数

```
5 5
s####
.####
.####
.####
....e
```

```
#include <iostream>
#include <string>
using namespace std;
int n,m;
string mp[15];
bool vis[15][15];
bool flag = false;
int cnt = 0;
int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

void dfs(int x, int y)
{
    if (vis[x][y])  return;
    vis[x][y] = true;
    if (mp[x][y] == 'e')
    {
        cnt++;
        return;
    }
    for (int i = 0; i < 4; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <n && ny >= 0 && ny < m && mp[nx][ny] !='#' && !v
        {
            dfs(nx,ny);
        }
    }
```

```
    vis[x][y] = false;
}

int main()
{
    int x,y;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        cin >> mp[i];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (mp[i][j] == 's' && !vis[i][j])
            {
                dfs(i,j);
            }
        }
    }
    cout << cnt <<endl;

    return 0;
}
```

6、最大的蛋糕块

```
5 6
.#....
..#...
..#..#
...##.
.#....
```

```
#include <iostream>
#include <string>
using namespace std;
int n,m,ans;
string mp[1005];
bool vis[1005][1005];
bool flag = false;
int cnt = 0;
int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

void dfs(int x, int y)
{
    if (vis[x][y])  return;
    vis[x][y] = true;
```

```
    cnt++;
    if (mp[x][y] == 'e')
    {
        cnt++;
        return;
    }
    for (int i = 0; i < 4; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <n && ny >= 0 && ny < m && mp[nx][ny] !='.' && !v
        {
            dfs(nx,ny);
        }
    }
}

int main()
{
    int x,y;
    cin >> n >> m;
    for (int i = 0; i < n; i++)
    {
        cin >> mp[i];
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            if (mp[i][j] == '#' && !vis[i][j])
            {
                cnt = 0;
                dfs(i,j);
                if (cnt > ans)
                    ans =cnt;
            }
        }
    }
    cout << ans <<endl;

    return 0;
}
```

7、家谱?

输出 n 行，每行有一个整数，表示第 i 个人有多少个直系后代。

样例输入 复制

```
4
1 2
1 3
2 4
```

样例输出 复制

```
3
1
0
0
```

```cpp
#include <iostream>
#include <string>
#include <vector>
using namespace std;
int n;
vector<int> son[100005];
bool f[100005];
int ans[100005];
int dx[4] = {1,0,-1,0};
int dy[4] = {0,-1,0,1};

int dfs(int u)
{
    int ret = 0;
    for (int i = 0; i < son[u].size(); i++)
    {
        ret+= dfs(son[u][i]);
    }
    ans[u] = ret;
    return ret + 1;
}

int main()
{
    int x,y,u;
    cin >> n ;
    for (int i = 0; i < n-1; i++)
    {
        cin >> x >> y;
        son[x].push_back(y);
        f[y] = true;
    }
    for (int i = 1; i <= n; i++)
    {
        if (!f[i])
        {
            u = i;
            break;
        }
    }
    dfs(u);
```

```
    for (int i = 1; i <= n; i++)
    {
        printf("%d\n",ans[i]);
    }

    return 0;
}
```

8、马的覆盖点

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;
int n,m;
char s[105][105];
int dx[8] = {2,2,1,1,-1,-1,-2,-2};
int dy[8] = {1,-1,2,-2,2,-2,1,-1};

void dfs(int x, int y,int step)
{
    if (step > 3)
        return;
    for (int i = 0; i < 8; i++)
    {
        int nx = x + dx[i];
        int ny = y + dy[i];
        if (nx >= 0 && nx <n && ny >= 0 && ny < m && s[nx][ny] =='.' )
        {
            s[x][y] = '#';
            dfs(nx,ny,step+1);
        }
    }
}

int main()
{
    int x,y;
    cin >> n >> m >> x >> y;

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            s[i][j] = '.';

        }
    }
    dfs(x-1,y-1,0);
    for (int i = 0; i < n; i++)
    {
```

```
42          printf("%s\n",s[i]);
43      }
44
45
46      return 0;
47  }
```