

Oj. 图的深搜

2018年12月14日 星期五 下午4:54

1233. Path

Description

大家还记得邻接表类吗？（如果你已经不记得，请看书上的P343页开始的内容）没错，邻接表是表示稀疏图（边数比较少的图）的一种很好的数据结构。

现在，我们要求使用深度优先遍历（若对深度优先搜索DFS的内容不清楚，请看P348）的思想，利用邻接表类，对给定的有向图，找出从指定结点start出发，长度为M的所有简单路径（简单路径是顶点序列中顶点不重复出现的路径）的数量。

为简化题目，我们还是约定：用正整数1,2,3.....n来表示每个结点的ID（编号）。(输入可能有重边)

Input Format

第1行：n m start M //正整数n，代表图中结点的数量。非负整数m代表要图中有向边的数量。start为起点编号，M为题中的简单路径长度

第2行到第1+m行: a b //每行两个整数：代表结点a到结点b有一条有向边（a->b），权值为1

Output Format

一个整数k，代表长度为M的所有简单路径的数量

Sample Input1

5 4 1 2

1 2

2 3

3 4

4 5

Sample Output1

1 //从1出发，长度为2的简单路径只有一条：1->2->3

Sample Input2

6 5 1 2

1 2

2 3

2 4

2 1

5 6

Sample Output2

2 //从1出发，长度为2的简单路径有2条：1->2->3,1->2->4。1->2->1不是，因为不是简单路径。

Limits

$0 < n, M \leq 10$ $0 \leq m \leq 100$

数据保证合法

(PS：关于简单路径的起点终点是否能相同，这个有点争议。本题规定不能相同)

思路：

用邻接表

深搜

记录路径长度

满足时总路径数加1

深搜

```
dfs(搜的点, 长度){
    将该点标记为已搜过
    if(长度到了){
        path++
        return
    }
    else{
        case1{
            如果没有visit过
            dfs(case1, 长度+1);
        }
        case2
```

```
case2 ...
```

```
case3 ...
```

```
}
```

```
    标记这个点没有visit过
```

```
}
```