

# 贝琪的学习任务【1.26】

## 任务简介：

从今天开始，你和渣澜的学习进度就一样啦

她比你多做了一次限时训练，你也比她多做了几道题

那今天的任务就是 python 的复习和算法的作业啦

如果做的不错的话，姐觉得可以安排让你踢渣澜的馆啦

(顺便再搞一次好玩的下注游戏)

## Python：

完成洛谷题目 1739、1009（用 Python 3）



## 算法：

今天主要学 DFS、BFS

深搜和广搜是两种很常见的搜索方式，属于图的遍历算法

深搜选择一个没有被搜过的结点（一般选择顶点），**按照深度优先**，一直往该结点的后续路径结点进行访问，直到该路径的最后一

个结点，然后再从未被访问的邻结点进行深度优先搜索，重复以上过程，直至所有点都被访问，遍历结束。

广搜以**广度优先**进行搜索。简言之就是先访问图的顶点，然后广度优先访问其邻接点，然后再依次进行被访问点的邻接点，一层一层访问，直至访问完所有点，遍历结束。

深搜一般以递归或者用栈进行深度搜索，而广搜一般配合队列进行搜索

深搜的典型例子比如树的前中后序遍历呀，广搜的例子比如树的层次遍历呀

下面就来看这个简单的例子：渣澜 23 号限时训练的最后一题：

题号：5318

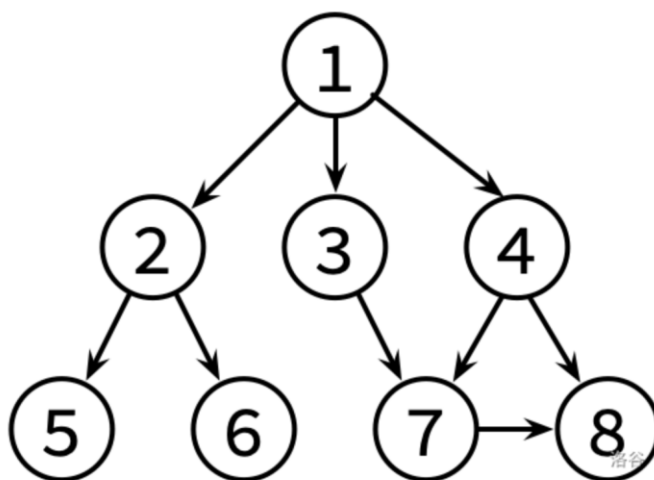
#### 题目描述

[展开](#)

小K 喜欢翻看洛谷博客获取知识。每篇文章可能会有若干个（也有可能没有）参考文献的链接指向别的博客文章。小K 求知欲旺盛，如果他看了某篇文章，那么他一定会去看这篇文章的参考文献（如果他之前已经看过这篇参考文献的话就不用再看它了）。

假设洛谷博客里面一共有  $n(n \leq 10^5)$  篇文章（编号为 1 到  $n$ ）以及  $m(m \leq 10^6)$  条参考文献引用关系。目前小 K 已经打开了编号为 1 的一篇文章，请帮助小 K 设计一种方法，使小 K 可以不重复、不遗漏的看完所有他能看到的文章。

这边是已经整理好的参考文献关系图，其中，文献  $X \rightarrow Y$  表示文章 X 有参考文献 Y。保证编号为 1 的文章没有被其他文章引用。



请对这个图分别进行 DFS 和 BFS，并输出遍历结果。如果有很多篇文章可以参阅，请先看编号较小的那篇（因此你可能需要先排序）。

## 输入输出样例

输入 #1

```
8 9
1 2
1 3
1 4
2 5
2 6
3 7
4 7
4 8
7 8
```

复制

输出 #1

```
1 2 5 6 3 7 8 4
1 2 3 4 5 6 7 8
```

复制

DFS 的话我们使用递归进行遍历，先输出根节点，之后对所有的子节点进行分析：如果没访问过，则递归遍历该子节点及其子树

```
void dfs(node* root){
    cout<<root->n<<" ";
    b[root->n-1] = 1;
    for(int i = 0;i<root->v.size();i++){
        if(b[root->v[i]->n-1] == 0) dfs(root->v[i]);
    }
}
```

BFS 的话我们用一个队列，先把根节点放进去，之后取出队列的头元素，输出该节点，然后遍历其子节点，如果没访问过的就放进队列里，然后再对队列头元素进行同样的操作，直到队列为空

```
void bfs(node* root){
    queue<node*> q;
    q.push(root);
    while(!q.empty()){
        node* t = q.front();
        q.pop();
        if(b[t->n-1] == 0) cout<<t->n<<" ";
        b[t->n-1] = 1;
        for(int i=0;i<t->v.size();i++)
            if(b[t->v[i]->n-1] == 0) q.push(t->v[i]);
    }
}
```

(我这话说着像一棵多叉树，但实际上这是一个图

可以自己尝试做一下这题，要姐的全代码可以直接说

**任务：**

算法：完成洛谷题目 1101、1443、1162、1605

有道题用深搜很好做

有道题用广搜很好做

有道题深搜广搜都好做

有道题暴力更好做

（其实这 4 道题都不好做

提交：把 4 道题的代码和 AC 截图**以及你选择的搜索方式**发给姐

就好啦

ddl：今晚 11.