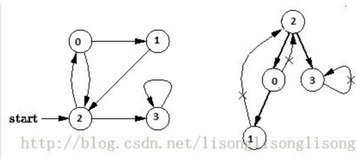
DFS && BFS

基本上是图论的基础基础。

1. DFS(深度优先搜索)

深度优先搜索的主要思想就是：从图中的某个点出发，找出其所有的邻接点中一个没被访问的点，然后再从这个邻接点开始访问他的邻接点中第一个没被访问的点。依此类推，直到图中所有的点都被访问。
值得注意的是，上面那种情况只适用于单个连通图，如果给出的图存在多个连通图，那么还必须在最初的出发点外层套一个循环，以避免有连通图没被访问。



从上图来进行DFS的话，最终输出结果是2，0，1，3.

DFS的代码实现:(图用邻接表表示)

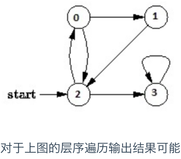
```
public class Graph {
    int V;//顶点数
    List<List<Integer>> adjs;//邻接表

    public Graph(int V) {
        this.V = V;
        adjs = new ArrayList<>(V);
        for (int i = 0; i < V; i++) {
            adjs.get(i) = new ArrayList<>();
        }
    }

    public void addEdge(int v, int w) {
        adjs.get(v).add(w);//有向图
        adjs.get(w).add(v);//无向图
    }

    public void dfs() {
        boolean[] visited = new boolean[V];

        for (int i = 0; i < V; i++) {//为了避免存在多个连通图
            if (!visited[i]) {
                DFSHelper(i, visited);
            }
        }
    }

    public void DFSHelper(int start, boolean[] visited) {
        visited[start] = true;
        List<Integer> adj = adjs.get(start);
        System.out.println(start);
        for (Integer i : adj) {
            if (!visited[i]) {
                DFSHelper(i, visited);
            }
        }
    }
}
```

比较简单。

- 空间复杂度：对于DFS而言，其空间复杂度就是栈的深度，而在最坏的情况下，栈的深度和顶点数相同，因此空间复杂度是$O(|V|)$
- 时间复杂度：DFS不仅要遍历图中顶点个数，而且边的个数也至少访问一遍，因此时间复杂度是$O(|V|+|E|)$.这是邻接表的情况。对于邻接矩阵的话，因为要遍历整个矩阵，因此时间复杂度是$O(|V|^2)$

2: BFS(广度优先搜索)

广度优先搜索的方式和DFS不同。BFS是找到一个点的某个未访问邻接点，然后继续访问下去；而对于BFS来说，是把某个点的所有邻接点都找出来，先遍历这些邻接点然后再访问这些邻接点的邻接点，显然这里用到的数据结构就是队列。广度优先搜索的另种叫法叫做层序遍历。



对于上图的层序遍历输出结果可能是2，0，3，1.

看代码:

```
public class Graph {
    int V;//顶点数
    List<List<Integer>> adjs;//邻接表

    public Graph(int V) {
        this.V = V;
        adjs = new ArrayList<>(V);
        for (int i = 0; i < V; i++) {
            adjs.get(i) = new ArrayList<>();
        }
    }

    public void addEdge(int v, int w) {
        adjs.get(v).add(w);//有向图
        adjs.get(w).add(v);//无向图
    }

    public void bfs() {
        boolean[] visited = new boolean[V];

        for (int i = 0; i < V; i++) {//为了避免存在多个连通图
            if (!visited[i]) {
                BFSHelper(i, visited);
            }
        }
    }
    public void BFSHelper(int start, boolean[] visited) {
        visited[start] = true;
        Deque<Integer> queue = new LinkedList<>();
        queue.offer(start);

        while (queue.size() > 0) {
            int cur = queue.poll();
            System.out.println(cur);
            List<Integer> adj = adjs.get(cur);
            for (Integer i : adj) {
                if (!visited[i]) {
                    queue.offer(i);
                }
            }
        }

    }
}
```

很简单的实现。

- 空间复杂度：对于BFS而言，其空间复杂度就是队列的空间，而在最坏的情况下，队列长度和顶点数相同，因此空间复杂度是$O(|V|)$
- 时间复杂度：和DFS一样，BFS不仅要遍历图中顶点个数，而且边的个数也至少访问一遍，因此时间复杂度是$O(|V|+|E|)$ 。这是邻接表的情况。对于邻接矩阵的话，因为要遍历整个矩阵，因此时间复杂度是$O(|V|^2)$