

模版·DFS(求路径数)

```
public class DepthFirstSearch{
    private int V;
    private LinkedList<Integer>[] adj;

    DepthFirstSearch(int v){
        V=v;
        adj=new LinkedList[V];
        for(int i=0;i<V;++i)
            adj[i]=new LinkedList();
    }

    void addEdge(int v,int w){
        adj[v].add(w);
    }

    void DFS(int v,boolean[] visited){
        visited[v]=true;
        System.out.print(v+"");

        Iterator<Integer> i=adj[v].listIterator();
        while(i.hasNext()){
            int n=i.next();
            if(!visited[n])
                DFS(n,visited);
        }
    }

    void DFS(int v){
        boolean[] visited=new boolean[V];
        DFS(v,visited);
    }

    public static void main(String args[]){
```

```
DepthFirstSearch dfs=new DepthFirstSearch(4);

dfs.addEdge(0,1);
dfs.addEdge(0,2);
dfs.addEdge(1,2);
dfs.addEdge(2,0);
dfs.addEdge(2,3);
dfs.addEdge(3,3);

System.out.println("从起点2开始遍历:");
dfs.DFS(2);
    }
}
```