

西南石油大学课程设计报告

课程	姓名	学号
算法分析与设计		
专业班级	任课老师	成绩
软件工程	王欣	

一、设计目的

图的广度优先搜索遍历近似于树的按层次遍历的过程。广度优先搜索遍历的特点是：尽可能先对横向进行搜索。设 x 和 y 是两个相继被访问过的顶点，若当前以 x 为出发点进行搜索，则在访问 x 的所有未曾被访问过的邻接点之后，紧接着以 y 为出发点进行横向搜索，并对搜索到的 y 的邻接点中尚未被访问的顶点进行访问，即先访问的顶点其邻接点亦先被访问。因此，算法实现需要引进队列保存已经被访问过的顶点。

实验目的是掌握BFS算法的代码实现。

二、设计内容

本次实验用邻接矩阵存储的方式表示图，借助辅助队列，实现广度优先搜索算法。将队列改为栈可实现DFS。

三、步骤

1. 首先定义一个访问标志数组 $visited[v]$ ，若值为 $true$ 则表示当前顶点 v 已被访问，初始值为 $false$ 。
2. 给定一个起始的顶点 v ，访问 v 并置该顶点的访问标志数组的值为 $true$ ，然后该顶点入队。
3. 只要队列不为空，则循环执行下列操作：

- a.队头元素 u 出队

b.依次检查 u 的所有邻接点 w ，如果 $visited[w]$ 的值为 $false$ ，则访问 w ，并置 $visited[w]$ 的值为 $true$ ，然后使 w 入队

注意，如果是非连通图，则在上述算法执行后一定还有顶点未被访问，需要从图中另选一个未被访问的顶点作为起始点，重复上述广度优先搜索过程，直到图中所有顶点均被访问过为止。

四、代码

```
void BFS(Graph G, int v)
{
    sqQueue Q;
    ArcType u;

    // 访问第v个顶点，并置访问标志数组相应分量值为true
    Visit(G.vexs[v]);
    visited[v] = true;

    // 辅助队列Q初始化，置空
    InitQueue(Q);
    // v进队
    EnQueue(Q, v);

    // 队列非空
    while (!QueueEmpty(Q))
    {
        // 队头元素出队并置为u
        DeQueue(Q, u);

        // 依次检查u的所有邻接点w，FirstAdjVex(G, u)表示u的第一个邻接点
        // NextAdjVex(G, u, w)表示u相对于w的下一个邻接点，w≥0表示存在邻接点
        for (ArcType w = FirstAdjVex(G, u); w ≥ 0; w = NextAdjVex(G, u, w))
        {
            // w为u的尚未访问的邻接顶点
            if (!visited[w])
            {
                // 访问顶点w
                Visit(G.vexs[w]);
                visited[w] = true; // 置访问标志数组相应分量值为true
                EnQueue(Q, w);    // w进队
            }
        }
    }
}
```

至于DFS算法，伪码如下：

```

DFS-A(G,s)
  for all v in V[G] do
    visited[v] := false
  end for
  S := EmptyStack
  Push(S,s)
  while not Empty(S) do
    u := Pop(S)
    if not visited[u] then
      visited[u] := true
      for all w in Adj[u] do
        if not visited[w] then
          Push(S,w)
        end if
      end for
    end if
  end while

```

DFS伪代码

可以看到，除了队列换成栈之外，其余的代码逻辑跟BFS算法是相同的。

五、体会

首先是确定图的表示结构时，这里使用邻接矩阵（而不采用邻接表）表示，因为较为方便，虽然可能会有空间的浪费，但这不并不是本次实验的重点，然后编写相关寻找邻接点的函数，便于BFS算法中搜索未被访问的顶点，同时用一个数组来记录各个结点的访问情况，最后用一个队列存储/输出相应的顶点。

在实验过程中，加深了对BFS算法的理解，同时对程序的魔力感到神奇，看上去和BFS算法大有不同的DFS算法，仅仅需要修改一处（队列换成栈）就能实现。