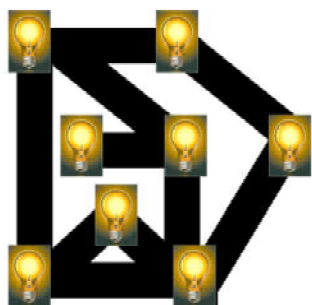


图的遍历（深度，广度）

2019年7月1日 15:25

深度优先DFS

深度优先搜索 (Depth First Search, DFS)



类似于树的先序遍历

```
void DFS ( Vertex V )
{ visited[ V ] = true;
  for ( V 的每个邻接点 W )
    if ( !visited[ W ] )
      DFS( W );
}
```

若有 N 个顶点、 E 条边，时间复杂度是

- 用邻接表存储图，有 $O(N+E)$
- 用邻接矩阵存储图，有 $O(N^2)$

代码：

/* 邻接表存储的图 - DFS */

```
void Visit(Vertex V)
```

```
{
```

```
    printf("正在访问顶点%d\n", V);
```

```
}
```

/* Visited[]为全局变量，已经初始化为false */

```
void DFS(LGraph Graph, Vertex V, void (*Visit)(Vertex))
```

```
{ /* 以V为出发点对邻接表存储的图Graph进行DFS搜索 */
```

```
    PtrToAdjVNode W;
```

```
    Visit(V); /* 访问第V个顶点 */
```

```
    Visited[V] = true; /* 标记V已访问 */
```

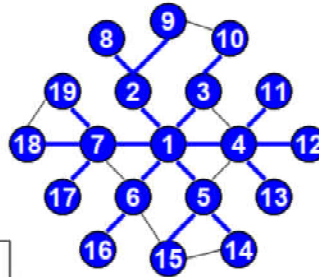
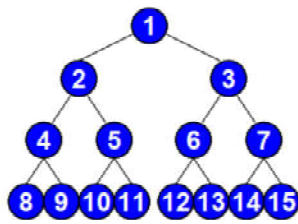
```
    for (W = Graph->G[V].FirstEdge; W; W = W->Next) /* 对V的每个邻接点W->AdjV */
```

```
        if (!Visited[W->AdjV]) /* 若W->AdjV未被访问 */
```

```
            DFS(Graph, W->AdjV, Visit); /* 则递归访问之 */
```

```
}
```

广度优先搜索(Breadth First Search, BFS)



```
void BFS ( Vertex V )
{ visited[V] = true;
  Enqueue(V, Q);
  while(!IsEmpty(Q)){
    V = Dequeue(Q);
    for ( v 的每个邻接点 W )
      if ( !visited[W] ) {
        visited[W] = true;
        Enqueue(W, Q);
      }
  }
}
```

若有N个顶点、E条边，时间复杂度是

- 用邻接表存储图，有 $O(N+E)$
- 用邻接矩阵存储图，有 $O(N^2)$

Copyright © 2014, 浙江大学计算机科学与技术学院

代码：

/* 邻接矩阵存储的图 - BFS */

/* IsEdge(Graph, V, W)检查<V, W>是否图Graph中的一条边，即W是否V的邻接点。
*/

/* 此函数根据图的不同类型要做不同的实现，关键取决于对不存在的边的表示方法。*/

/* 例如对有权图, 如果不存在边被初始化为INFINITY, 则函数实现如下: */

```
bool IsEdge(MGraph Graph, Vertex V, Vertex W)
{
    return Graph->G[V][W] < INFINITY ? true : false;
}
```

/* Visited[]为全局变量，已经初始化为false */

void BFS(MGraph Graph, Vertex S, void (*Visit)(Vertex))

{ /* 以S为出发点对邻接矩阵存储的图Graph进行BFS搜索 */

Queue Q;

Vertex V, W;

Q = CreateQueue(MaxSize); /* 创建空队列, MaxSize为外部定义的常数 */

/* 访问顶点S: 此处可根据具体访问需要改写 */

Visit(S);

Visited[S] = true; /* 标记S已访问 */

AddQ(Q, S); /* S入队列 */

while (!IsEmpty(Q)) {

V = DeleteQ(Q); /* 弹出V */

for (W = 0; W < Graph->Nv; W++) /* 对图中的每个顶点W */

/* 若W是V的邻接点并且未访问过 */

if (!Visited[W] && IsEdge(Graph, V, W)) {

/* 访问顶点W */

Visit(W);

Visited[W] = true; /* 标记W已访问 */

AddQ(Q, W); /* W入队列 */

}

```
    } /* while结束 */  
}
```

图不连通怎么办？

```
void DFS ( Vertex V )  
{ visited[ V ] = true;  
  for ( V 的每个邻接点 W )  
    if ( !visited[ W ] )  
      DFS( W );  
}
```

每调用一次DFS(V)，就把V所在的连通分量遍历了一遍。BFS也是一样。

```
void ListComponents ( Graph G )  
{ for ( each V in G )  
  if ( !visited[V] ) {  
    DFS( V ); /*or BFS( V )*/  
  }  
}
```