

# 图相关

2024年12月30日 星期一 21:30

## 1. BFS

```
BFS(v) {  
    把v标记为已到达  
    初始化队列Q, 放入v  
    while (!Q.empty()) {  
        从Q中取出w  
        令u为与w邻接的点  
        while (u != NULL) {  
            if (u未被标记) {  
                将u入队  
                把u标记为到达  
            }  
            u = w的下一个邻接点  
        }  
    }  
}
```

## 2. DFS

```
DFS(v) {  
    把v标记为已到达  
    for (v的每个邻接点u) {  
        DFS(u);  
    }  
}
```

## 3. 拓扑排序

```
令n表示有向图顶点数  
令theOrder是空序列  
while (true) {  
    令w是任意一个没有入边(v,w)的顶点, 其中v不在theOrder中  
    如果没有这样的顶点w, 程序终止  
    把w加到theOrder尾部  
}  
if (theOrder 顶点数 < n)  
    算法失败  
else  
    theOrder 是一个拓扑序列
```

## 4. Kruskal

```
令T是选定的边集, 初始时T=∅  
令E是网络的边集  
while (E ≠ ∅) && (|T| ≠ n-1) {  
    令(u,v)是E中一条成本最小的边  
    E = E - {(u,v)}  
    if ((u,v)在T中不会产生环路)  
        把(u,v)加入T  
}  
if (|T| == n-1)  
    T是一棵最小成本生成树  
else  
    不连通, 无生成树
```

## 5. Prim

```
令T是入选的边集, 初始化T=∅  
令TV是已在树中的顶点集, 置TV={1}  
令E是网络的边集  
while (E ≠ ∅) && (|TV| ≠ n-1) {  
    令(u,v)是一条成本最小的边, 且u ∈ TV, v ∉ TV  
    if (没有这样的边)  
        终止循环  
    E = E - {(u,v)}  
    把(u,v)加入T  
    把v加入TV  
}  
if (|TV| == n-1) {  
    T是一棵最小成本生成树  
} else {  
    不连通, 无生成树  
}
```