

# 图的应用

---

生成树,

## 生成树

连通图的生成树是一个极小连通子图（包含全部的点，和刚好足够连通的边）：

有 $n$ 个顶点的生成树仅有 $n-1$ 条边；

若 $<n-1$ ，则是非连通图；

$>n-1$ 则一定有环；

有 $n-1$ 条边不一定是生成树。

一种典型的应用是城市间修路（点表示城市、边表示路、边的权值表示距离， $n$ 个城市间可修 $n(n-1)/2$ 条路，选出其中 $n-1$ 条使得总路程最短（权值最小））。其基本思想为：

- 选择 $n-1$ 条边构成最小生成树；
- 尽可能选权值最小的边，且不构成回路。

## 最小生成树MST算法

有两种算法：普利姆Prim算法（一种贪心算法，贪吃蛇），克鲁斯卡尔Kruskal算法（Prim的延申，相对更为符合最小生成树思想）

### 普利姆Prim算法

从 $v_0$ 出发，先找权值最小的边，并且不构成环，找到后则构成一个整体，继续找对于这个整体而言权值最小的边，一直重复直到连通所有的点，即为最小生成树。

时间复杂度为 $O(n^2)$ ，与边数目无关。

### 克鲁斯卡尔Kruskal算法

将边的权值排序，每次选取最小边（前提不构成回路），直到选取完 $n-1$ 条边。

数组初始化的时间复杂度为 $O(n)$ ；排序权值采用堆排序或快速排序，时间复杂度为 $O(e \log e)$ 。

克鲁斯卡尔算法更适用于稀疏图（较多点的度为1）。

## MST唯一性讨论

1. 权重均不相等，MST必定唯一；
2. 存在环，且环上有相等权重，同时相等权重比别的全重大，MST不唯一——但权值之和是唯一的。

## 最短路径

从一个点到另一个点，所经过的边的权值和最小。有迪杰斯特拉Dijkstra算法（贪心算法，路径按长度递增次序产生最短路径）和弗洛伊德Floyd算法（构建二维数组，记录任意两点的最短路径并逐步更新数组）。

### Dijkstra

从一点出发，求出到各个顶点的最短路径和路径长度，按长度递增次序生成个顶点的最短路径，直到求出长度最长的最短路径。

但由于是一种贪心算法，所以只关心局部最优解，且**不适用于负权值和负回路**。

### 步骤如下，采用做表的形式

1. 指定点后，初始化它到各个点的距离（若无法直接到达，则标记为 $\infty$ ），选取其中最小点作为下一个到的点；
2. 随后将该点加入，再求该整体到剩余点的距离（如果更近了（权值更小），则更新，并将其pre更新为该点），选出最近的点作为下一个点；
3. 将该点加入，重复2；
4. 直到所有的点遍历完成。

### 算法分析

- 初始化时间复杂度 $O(n)$ ；
- 求最短路径二重循环的时间复杂度 $O(n^2)$ 。

因此，整体的时间复杂度为 $O(n^2)$ 。

但以上只是从一点到其余点的时间复杂度。如果是求各个点到其余点的最短路径，时间复杂度将是 $O(n^3)$ 。

### Floyd

构建二维数组，记录最短长度，然后逐步更新数组中的数值，直到求出所有结点之间的最短路径。

适用于**稠密图**，且可以处理负权边（但不允许负回路），时间复杂度 $O(n^3)$ ；  
空间复杂度 $O(n^2)$ 。

### 步骤如下

1. 列出原图的邻接矩阵A，途径点矩阵Path（初始值均为-1），途径点数组S（初始化为空）；
2. 随后选一个点作为途径点，若经过该点可以获得更短的路径，则更新A上的权值；
3. 将该点加入S；
4. 对于更新权值的点，更新Path为当前途径点；
5. 重复2、3、4，直到S内点的数量等于图的数量；
6. 结合A和Path，Path中每一行的最大值，即为对应A上该点到另一点的最短路径。

## AOV网与拓扑排序【占题目的40%】

**AOV网**是顶点表示活动的有向无环图。

AOV网用于工程项目中的工序、工程时间进度的问题，是一种有向无环图，顶点表示活动，有向边表示活动间优先关系。

**拓扑排序**是AOV的遍历算法，用于确定活动执行顺序。

当一个有向无环图所组成的序列中：当每个顶点仅出现一次，且后继节点绝对没有通往前驱节点的路径，即可使用拓扑排序。

时间复杂度为 $O(n+e)$ 。

## 拓扑排序的过程

1. 选择一个没有前驱结点的顶点并输出；
2. 删除该点，以及从该店出发的所有弧；
3. 重复1、2，直到全部顶点输出（或图中不存在无前驱的顶点）。

## 一种快速排除AOV错误遍历的方式

若选项中出现了违逆弧指向的遍历，则为错误。

## AOE网与关键路径

**AOE网**是边表示活动的有向无环图，通常边的权值是活动时间。

**关键路径**是从起点到终点的最长路径（完成工程的最长时间，是影响整个工程的关键）。

关键活动是关键路径上的活动。增加关键活动，关键路径必然增长。

## 事件最早/最晚发生时间

1. 绘制表格，顶横为活动（边），顶列为最早发生时间、最晚发生时间、松弛量（最早、最晚发生事件的差值）。
2. 求最早发生时间，用正推法（从左到右），取权值和最大的那条路径；
3. 求最晚发生时间，用倒推法（从右到左），取权值和最小的哪条路径，并用关键路径的长度-该路径长度；
4. 求松弛量。

## 时间复杂度总结

$O(n+e)$

**邻接表**创建的：BFS、DFS、拓朴排序、AOE网，创建表、插入边、删除边。

$O(n^2)$

**邻接矩阵**创建的：BFS、DFS、拓朴排序、AOE网，创建表、插入边、删除边。

普利姆算法，迪杰斯特拉算法。

$O(e \log e)$

克鲁斯卡尔算法。

$O(n^3)$

弗洛伊德算法。