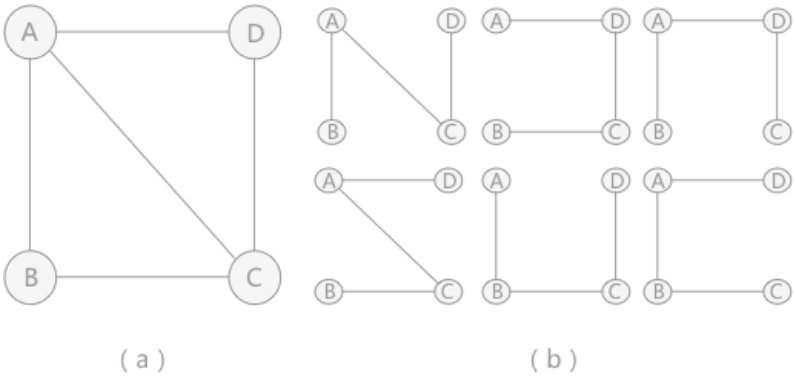
最小生成树

1. 什么是最小生成树

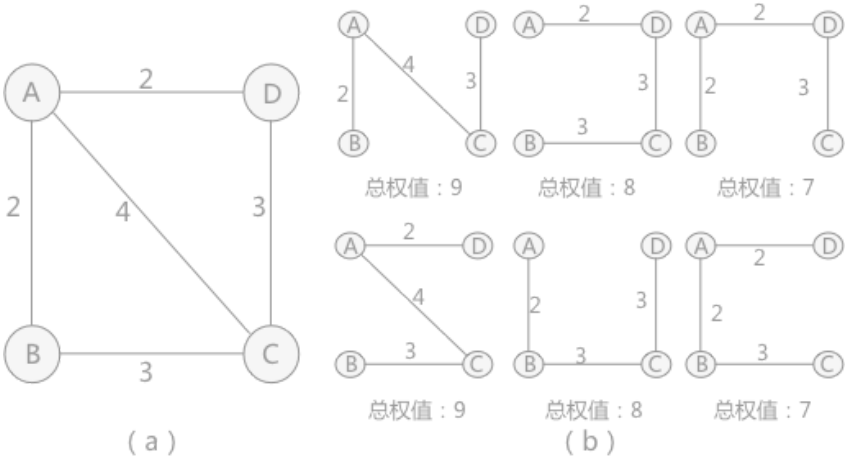
1.1 生成树的定义

一个连通图的生成树是一个极小的连通子图，它包含图中全部的n个顶点，但只有构成一棵树的n–1条边。



连通图和它相对应的生成树，可以用于解决实际生活中的问题：假设A、B、C 和 D 为 4 座城市，为了方便生产生活，要为这 4 座城市建立通信。对于 4 个城市来讲，本着节约经费的原则，只需要建立 3 个通信线路即可，就如图 （b）中的任意一种方式。

在具体选择采用（b）中哪一种方式时，需要综合考虑城市之间间隔的距离，建设通信线路的难度等各种因素，将这些因素综合起来用一个数值表示，当作这条线路的权值。



综合所看，选择权值总和为7的两种方案最节约经费。

1.2 生成树的属性

- 一个连通图可以有多个生成树；

- 一个连通图的所有生成树都包含相同的顶点个数和边数；
- 生成树当中不存在环；
- 移除生成树中的任意一条边都会导致图的不连通， 生成树的边最少特性；
- 在生成树中添加一条边会构成环。 对于包含n个顶点的连通图，生成树包含n个顶点和n−1条边；
- 对于包含n个顶点的无向完全图最多包含n^{n-2} 颗生成树。

1.3 最小生成树

所谓一个 带权图 的最小生成树，就是原图中边的权值最小的生成树 ，所谓最小是指边的权值之 和小于或者等于其它生成树的边的权值之和。

最小生成树的算法思想就是从顶点集或边集的角度来进行贪婪化。

2. 克鲁斯卡尔（Kruskal）算法

克鲁斯卡尔算法（Kruskal）是一种使用贪婪方法的最小生成树算法。该算法初始将图视为森林， 图中的每一个顶点视为一棵单独的树。一棵树只与它的邻接顶点中权值最小且不违反最小生成树属性 （不构成环）的树之间建立连边。

从边的角度求网的最小生成树，更适合于求边稀疏的网的最小生成树。

基本思想：按照权值从小到大的顺序选择n−1条边，并保证这n−1条边不构成回路。

具体做法：首先构造一个只含n个顶点的森林，然后依权值从小到大从连通网中选择边加入到森林中，并使森林中不产生回路，直至森林变成一棵树为止。

代码编写的问题：

问题一 对图的所有边按照权值大小进行排序。

问题二 将边添加到最小生成树中时，怎么样判断是否形成了回路。

克鲁斯卡尔算法的具体思路是：

将所有边按照权值的大小进行升序排序，然后从小到大一一判断，条件为：

如果这个边不会与之前选择的所有边组成回路，就可以作为最小生成树的一部分；反之，舍去。

直到具有 n 个顶点的连通网筛选出来 n−1 条边为止。筛选出来的边和所有的顶点构成此连通网的最小生成树。

> 判断是否会产生回路的方法为：在初始状态下给每个顶点赋予不同的标记，对于遍历过程的每条边，其都有两个顶点，判断这两个顶点的标记是否一致，如果一致，说明它们本身就处在一棵树中，如果继续连接就会产生回路；如果不一致，说明它们之间还没有任何关系，可以连接。

3. 贪婪算法

3.1 贪心算法思想

贪心算法总是作出在当前看来最好的选择。也就是说贪心算法并不从整体最优考虑， 它所作出的选择只是在某种意义上的局部最优选择。当然，希望贪心算法得到的最终结果也是整体最优 的。虽然贪心算法不能对所有问题都得到整体最优解，但对许多问题它能产生整体最优解。如单源最短 路经问题，最小生成树问题等。在一些情况下，即使贪心算法不能得到整体最优解，其最终结果却是最 优解的很好近似。

贪心算法的基本要素：

1. 贪心选择性质。所谓贪心选择性质是指所求问题的整体最优解可以通过一系列局部最优的选 择，即贪心选择来达到。这是贪心算法可行的第一个基本要素，也是贪心算法与动态规划算法的主要区别。

动态规划算法通常以自底向上的方式解各子问题（后面会讲），而贪心算法则通常以自顶向下的 方式进行，以迭代的方式作出相继的贪心选择，每作一次贪心选择就将所求问题简化为规模更小的子问题。

对于一个具体问题，要确定它是否具有贪心选择性质，必须证明每一步所作的贪心选择最终导致 问题的整体最优解。

2. 当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。问题的最优子 结构性质是该问题可用动态规划算法或贪心算法求解的关键特征。

3.2 贪心算法的基本思路

从问题的某一个初始解出发逐步逼近给定的目标，以尽可能快的地求得更好的解。当达到算法中的某一步不能再继续前进时，算法停止。

该算法存在问题：

1、不能保证求得的最后解是最佳的；

2、不能用来求最大或最小解问题；

3、只能求满足某些约束条件的可行解的范围。

实现该算法的过程：

从问题的某一初始解出发；

while 能朝给定总目标前进一步 do

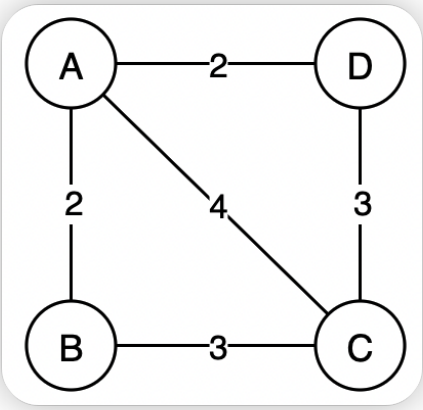
求出可行解的一个解元素；

由所有解元素组合成问题的一个可行解。

4. 普里姆（Prim）算法

普里姆算法在找最小生成树时，将顶点分为两类，一类是在查找的过程中已经包含在生成树中的 顶点（假设为 A 类），剩下的为另一类（假设为 B 类）。

对于给定的连通网，起始状态全部顶点都归为 B 类。在找最小生成树时，选定任意一个顶点作为 起始点，并将之从 B 类移至 A 类；然后找出 B 类中到 A 类中的顶点之间权值最小的顶点，将之从 B 类移至 A 类，如此重复，直到 B 类中没有顶点为止。所走过的顶点和边就是该连通图的最小生成树。



4.1 举例说明

1. 假如从顶点A出发，顶点 B、C、D 到顶点 A 的权值分别为 2、4、2，所以，对于顶点 A 来说，顶点 B 和顶点 D 到 A 的权值最小，假设先找到的顶点 B；
2. 继续分析顶点 C 和 D，顶点 C 到 B 的权值为 3，到 A 的权值为 4；顶点 D 到 A 的权值为 2，到 B 的权值为无穷大（如果之间没有直接通路，设定权值为无穷大）。所以顶点 D 到 A 的权值最小；
3. 最后，只剩下顶点 C，到 A 的权值为 4，到 B 的权值和到 D 的权值一样大，为 3。所以该连通图有两个最小生成树；

5. 总结

最小生成树的问题，简单得理解就是给定一个带有权值的连通图（连通网），从众多的生成树中 筛选出权值总和最小的生成树，即为该图的最小生成树。

最经典的两个最小生成树算法：Kruskal 算法与 Prim 算法。两者分别从不同的角度构造最小生成树，Kruskal 算法从边的角度出发，使用贪心的方式选择出图中的最小生成树，而 Prim 算法从顶点的角 度出发，逐步找各个顶点上最小权值的边来构建最小生成树的。

最小生成树问题应用广泛，最直接的应用就是网线架设（网络G表示n各城市之间的通信线路网线 路（其中顶点表示城市，边表示两个城市之间的通信线路，边上的权值表示线路的长度或造价。可通过 求该网络的最小生成树达到求解通信线路或总代价最小的最佳方案。或者如果我们需要用最少的电线给 一所房子安装电路。）、道路铺设。还可以间接应用于纠错的LDPC码、Renyi 熵图像配准、学习用于实时脸部验证的显著特征、减少蛋白质氨基酸序列测序中的数据存储，在湍流（turbulent）中模拟粒子交互的局部性，以及用于以太网桥接的自动配置，以避免在网络中形成环路。除此之外，最小生成树在聚类算法中也是应用广泛。