**迪克斯特拉算法-- Dijkstra's Algorithm**

在图形应用中，常常需要求从图中某个结点至其余各结点的最短路径，如对于一个物流配送系统计算从配送中心到各订货点的最短路径。

Dijkstra's Algorithm 基本思想：

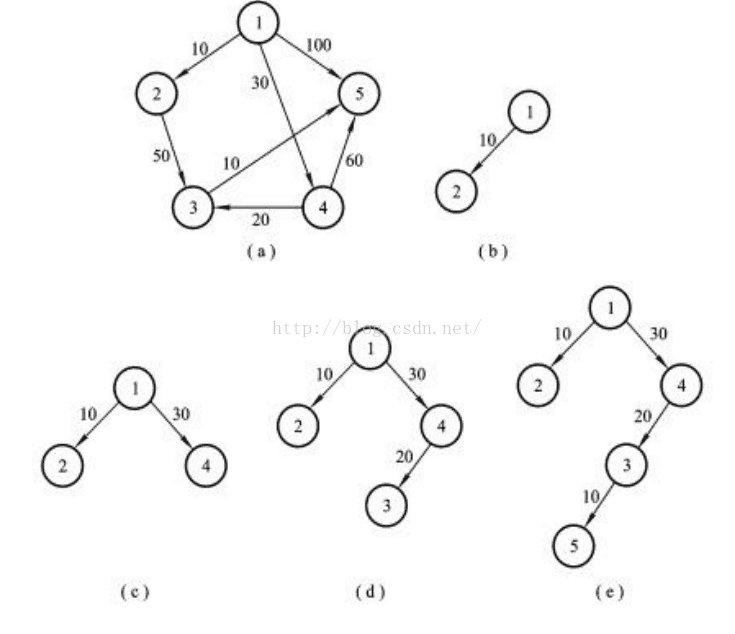
若给定带权有向图G=(V,E)和源顶点v0，构筑一个源集合S，将v0加入其中。

① 对差集V\S中个顶点vi，逐一计算从v0 至它的距离 D(v0, vi ),若该两顶点之间没有边，则其距离为无穷大。求出其中距离最短的顶点w，将其加入到集合 S 中。

② 重新计算 v0 至差集 V\S 中各顶点的距离 D（v0, vi）= Min(D(v0, vi ), D(v0, w ) + C(w, vi )).其中C（w, vi）是顶点w 与 vi 之间边上的费用。

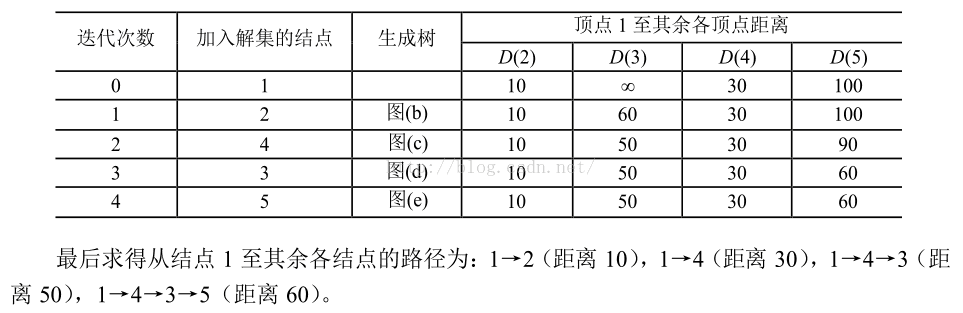
③ 重复 步骤①②。直至所有的顶点都加到集合S 中为止。

算法求解过程图式：



把该图看成是物流配送系统，边上的数字是各地间距离，配送中心位于结点1处，根据该算法就可以设计出从结点 1 至其他各个结点线路最短的配送线路。

步骤：



小结：

Dijkstra's 算法与最小生成树的区别在于：

① 最小生成树是对全图而言的，而Dijkstra's算法是对某个结点而言的。

② 最小生成树是连接所有结点的最短路径，但是如果从某个结点出发，沿着最小生成树到另一个结点的路径不一定是最短的。而在Dijkstra's树中，从根结点到各叶子结点的路径都是最短的。

③ 若Dijkstra's算法依次应用于每个顶点，最后可以得到任意两个顶点之间的最短路径，这就是通常所说的任意顶点对之间的最短路径问题（all-pairs shortest paths,APAP）

ps:Floyd算法也是求解这类问题的算法。

---------------------

作者：吴威龙

来源：CSDN

原文：https://blog.csdn.net/leaf\_130/article/details/50668868

版权声明：本文为博主原创文章，转载请附上博文链接！