Dijkstra算法

按路径长度递增次序产生算法：

把顶点集合V分成两组：

（1）S：已求出的顶点的集合（初始时只含有源点V0）

（2）V-S=T：尚未确定的顶点集合

将T中顶点按递增的次序加入到S中，保证：

（1）从源点V0到S中其他各顶点的长度都不大于从V0到T中任何顶点的最短路径长度

（2）每个顶点对应一个距离值

S中顶点：从V0到此顶点的长度

T中顶点：从V0到此顶点的只包括S中顶点作中间顶点的最短路径长度

依据：可以证明V0到T中顶点Vk的，或是从V0到Vk的直接路径的权值；或是从V0经S中顶点到Vk的路径权值之和

（[反证法](https://baike.baidu.com/item/%E5%8F%8D%E8%AF%81%E6%B3%95)可证）

求最短路径步骤

算法步骤如下：

G={V,E}

1. 初始时令 S={V0},T=V-S={其余顶点}，T中顶点对应的距离值

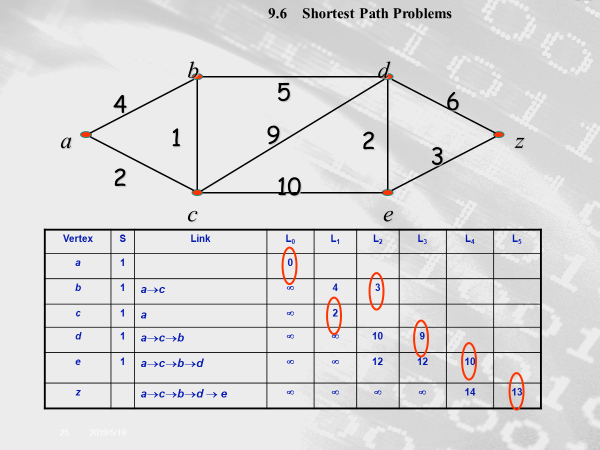
若存在<V0,Vi>，d(V0,Vi)为<V0,Vi>弧上的权值

若不存在<V0,Vi>，d(V0,Vi)为∞

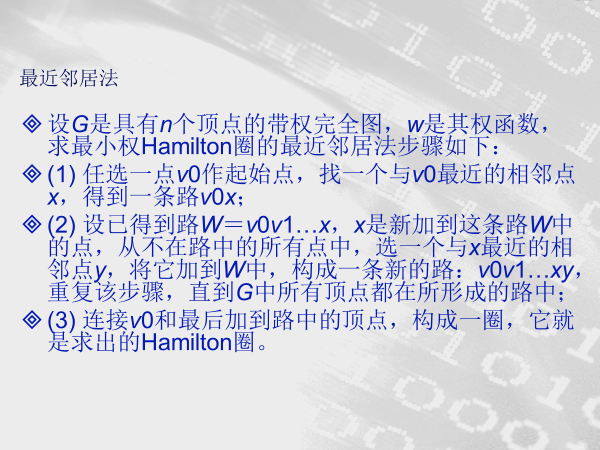
2. 从T中选取一个与S中顶点有关联边且权值最小的顶点W，加入到S中

3. 对其余T中顶点的距离值进行修改：若加进W作中间顶点，从V0到Vi的距离值缩短，则修改此距离值

重复上述步骤2、3，直到S中包含所有顶点，即W=Vi为止

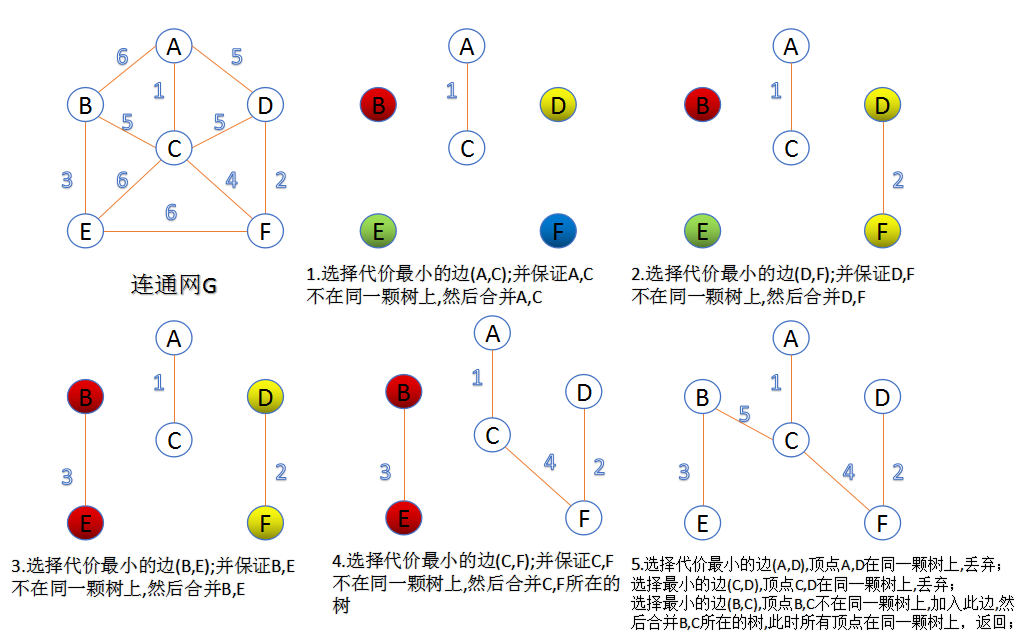


最近邻居法



Kurskral算法—最小生成树

此算法可以称为“加边法”，初始最小生成树边数为0，每迭代一次就选择一条满足条件的最小代价边，加入到最小生成树的边集合里。   
1. 把图中的所有边按代价从小到大排序；   
2. 把图中的n个顶点看成独立的n棵树组成的森林；   
3. 按权值从小到大选择边，所选的边连接的两个顶点ui,viui,vi,应属于两颗不同的树，则成为最小生成树的一条边，并将这两颗树合并作为一颗树。   
4. 重复(3),直到所有顶点都在一颗树内或者有n-1条边为止。



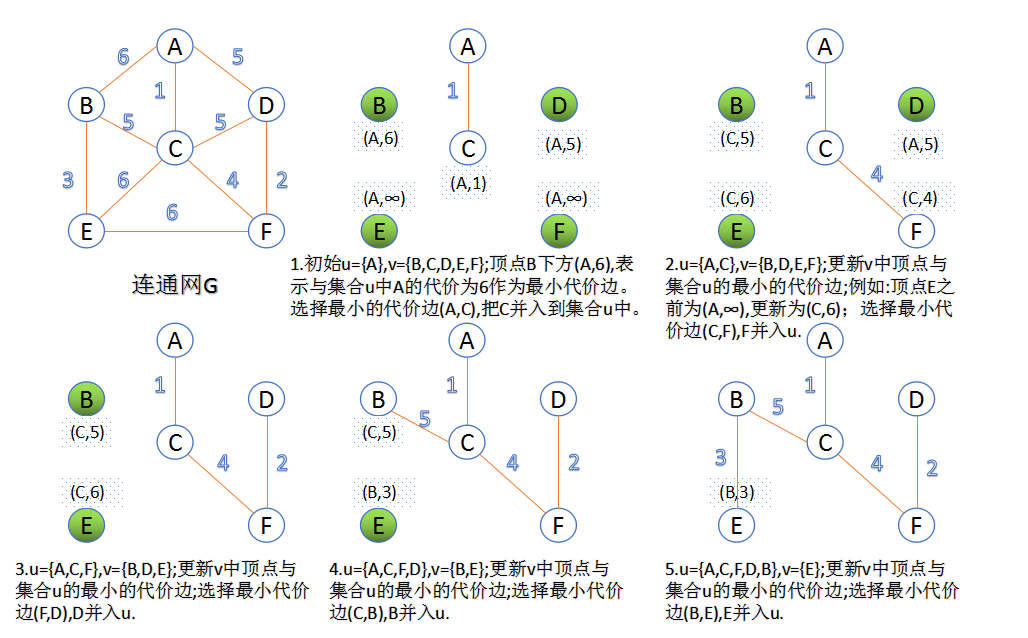
补充（Prim算法—最小生成树）

此算法可以称为“加点法”，每次迭代选择代价最小的边对应的点，加入到最小生成树中。算法从某一个顶点s开始，逐渐长大覆盖整个连通网的所有顶点。

图的所有顶点集合为VV；初始令集合u={s},v=V−uu={s},v=V−u;

在两个集合u,vu,v能够组成的边中，选择一条代价最小的边(u0,v0)(u0,v0)，加入到最小生成树中，并把v0v0并入到集合u中。

重复上述步骤，直到最小生成树有n-1条边或者n个顶点为止。



沃舍尔算法

沃舍尔算法，是一种特殊的求闭包的高效算法，其目的是构造一个传递的二元关系，因为避免了由矩阵乘法带来的大量的乘法运算，所以运用的比较广，（用它也可以来判断图的联通性），其实沃舍尔的这个算法证明，借用了图的一些性质（用到了关系矩阵，还可以由图来表示，书上应该也给出了讲解）；  
一个图由N个顶点组成，现在给定了一个二元关系（也可以说是一个函数，实质上是构造了映射关系），我们现在来看，设这些顶点依次为x1，x2......xn,如果该关系中有（xi，xj）则在两点间加一条有向边；设这个图为基图G；  
现在来看，图G中必然存在至少一组这样的关系，xi与xj有边,xj与xk有边,但是xi与xk没有边;否则该关系本身就是一个传递关系了;（不知道你现在有没有理解到，构造这个闭包实际上就是加边，如果有上述的情况，就在xi，xk间加一条边，加边和[矢量三角形](https://www.baidu.com/s?wd=%E7%9F%A2%E9%87%8F%E4%B8%89%E8%A7%92%E5%BD%A2&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)法则类似，但是加完边了，可能又会产生新的这样的情况，那么就要一直进行下去，继续加边，但是这样加边，到N此之后一定可以完成，这一点极其重要，你自己画个图试一下这个过程，就会很快明白为什么一定能行，这也是通过R+R^1....R^n求出闭包这种方法的依据所在；任何要加的边一定能在这个幂运算中产生）；但是对大矩阵这样做是不划算的；沃舍尔提供了一种不需要乘法的运算；首先我们理解最后的闭包，是把所有顶点需要加的边都加完了。这样才叫闭包，否则就不是闭包，那么现在来看，如果我加的边，构成的圈只经过了x1那么如何构造出只经过x1，x2的边呢，这样的边可以分为两种；只经过了x1的边肯定满足，经过x2的边可以看成<xi,x2><x2,xj>这是一种递归的思想

长度为k的路径数量

![C:\Users\sam\Documents\Tencent Files\2472076590\Image\C2C\~JX](0`$~(@AD$FS5GGX$JE.png](data:image/png;base64,)

