**Prim**[**算法**](http://lib.csdn.net/base/datastructure)

**1.概览**

**普里姆算法**（Prim算法），图论中的一种算法，可在加权连通图里搜索最小生成树。意即由此算法搜索到的边子集所构成的树中，不但包括了连通图里的所有顶点（英语：Vertex (graph theory)），且其所有边的权值之和亦为最小。该算法于1930年由捷克数学家沃伊捷赫·亚尔尼克（英语：Vojtěch Jarník）发现；并在1957年由美国计算机科学家罗伯特·普里姆（英语：Robert C. Prim）独立发现；1959年，艾兹格·迪科斯彻再次发现了该算法。因此，在某些场合，普里姆算法又被称为DJP算法、亚尔尼克算法或普里姆－亚尔尼克算法。

**2.算法简单描述**

1).输入：一个加权连通图，其中顶点集合为V，边集合为E；

2).初始化：Vnew = {x}，其中x为集合V中的任一节点（起始点），Enew = {},为空；

3).重复下列操作，直到Vnew = V：

a.在集合E中选取权值最小的边<u, v>，其中u为集合Vnew中的元素，而v不在Vnew集合当中，并且v∈V（如果存在有多条满足前述条件即具有相同权值的边，则可任意选取其中之一）；

b.将v加入集合Vnew中，将<u, v>边加入集合Enew中；

4).输出：使用集合Vnew和Enew来描述所得到的最小生成树。

下面对算法的图例描述

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **图例** | **说明** | **不可选** | **可选** | **已选（Vnew）** |
| http://pic002.cnblogs.com/images/2012/426620/2012073015154494.png | 此为原始的加权连通图。每条边一侧的数字代表其权值。 | - | - | - |
| http://pic002.cnblogs.com/images/2012/426620/2012073015175038.png | 顶点**D**被任意选为起始点。顶点**A**、**B**、**E**和**F**通过单条边与**D**相连。**A**是距离**D**最近的顶点，因此将**A**及对应边**AD**以高亮表示。 | C, G | A, B, E, F | D |
| http://pic002.cnblogs.com/images/2012/426620/2012073016090032.png | 下一个顶点为距离**D**或**A**最近的顶点。**B**距**D**为9，距**A**为7，**E**为15，**F**为6。因此，**F**距**D**或**A**最近，因此将顶点**F**与相应边**DF**以高亮表示。 | C, G | B, E, F | A, D |
| http://pic002.cnblogs.com/images/2012/426620/2012073016130394.png | 算法继续重复上面的步骤。距离**A**为7的顶点**B**被高亮表示。 | C | B, E, G | A, D, F |
| http://pic002.cnblogs.com/images/2012/426620/2012073016143177.png | 在当前情况下，可以在**C**、**E**与**G**间进行选择。**C**距**B**为8，**E**距**B**为7，**G**距**F**为11。**E**最近，因此将顶点**E**与相应边**BE**高亮表示。 | 无 | C, E, G | A, D, F, B |
| http://pic002.cnblogs.com/images/2012/426620/2012073016154616.png | 这里，可供选择的顶点只有**C**和**G**。**C**距**E**为5，**G**距**E**为9，故选取**C**，并与边**EC**一同高亮表示。 | 无 | C, G | A, D, F, B, E |
| http://pic002.cnblogs.com/images/2012/426620/2012073016114494.png | 顶点**G**是唯一剩下的顶点，它距**F**为11，距**E**为9，**E**最近，故高亮表示**G**及相应边**EG**。 | 无 | G | A, D, F, B, E, C |
| http://pic002.cnblogs.com/images/2012/426620/2012073016100874.png | 现在，所有顶点均已被选取，图中绿色部分即为连通图的最小生成树。在此例中，最小生成树的权值之和为39。 | 无 | 无 | A, D, F, B, E, C, G |

**3.简单证明prim算法**

反证法：假设prim生成的不是最小生成树

1).设prim生成的树为G0

2).假设存在Gmin使得cost(Gmin)<cost(G0)   则在Gmin中存在<u,v>不属于G0

3).将<u,v>加入G0中可得一个环，且<u,v>不是该环的最长边(这是因为<u,v>∈Gmin)

4).这与prim每次生成最短边矛盾

5).故假设不成立，命题得证.

**5.时间复杂度**

这里记顶点数v，边数e

邻接矩阵:O(v2)                 邻接表:O(elog2v)