Prim算法

设计思想

输入: 图 $G=(V,E,W), V=\{1,2,...,n\}$

输出: 最小生成树 T

设计思想:

初始 $S = \{1\}$,

选择连接 S = V - S 集合的最短边 $e = \{i,j\}$,其中 $i \in S, j \in V - S$. 将 e 加入树 T, j 加入 S.

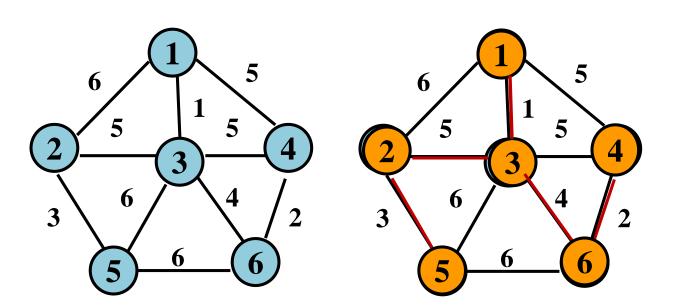
继续执行上述过程,直到 S=V 为止.

伪码

算法 Prim (G, E, W)

- 1. $S \leftarrow \{1\}$
- 2. while $V S \neq \emptyset$ do
- 4. $S \leftarrow S \cup \{j\}$

实例



正确性证明:归纳法

命题:对于任意 k < n,存在一棵最小生成树包含算法前 k 步选择的边.

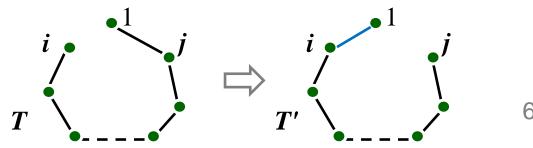
归纳基础: k = 1, 存在一棵最小生成树 T 包含边 $e = \{1, i\}$, 其中 $\{1, i\}$ 是所有关联 1 的边中权最小的.

归纳步骤: 假设算法前 k 步选择的边构成一棵最小生成树的边,则算法前 k+1 步选择的边也构成一棵最小生成树的边.

归纳基础

证明:存在一棵最小生成树 T 包含关联结点1的最小权的边 $e=\{1,i\}$.

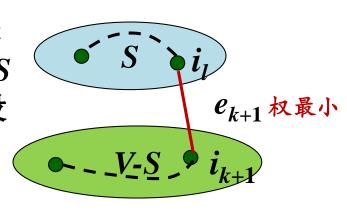
证 设 T 为一棵最小生成树,假设 T 不包含 $\{1,i\}$,则 $T \cup \{\{1,i\}\}$ 含有一条回路,回路中关联1的另一条边 $\{1,j\}$.用 $\{1,i\}$ 替换 $\{1,j\}$ 得到树 T',则 T' 也是生成树,且 $W(T') \leq W(T)$.



归纳步骤

假设算法进行了k步,生成树的边为 e_1, e_2, \ldots, e_k ,这些边的端点构成集合S. 由归纳假设存在G的一棵最小生成树T包含这些边.

算法第 k+1 步选择 顶点 i_{k+1} ,则 i_{k+1} 到S中顶点边权最小,设 此边 $e_{k+1}=\{i_{k+1},i_l\}$. 若 $e_{k+1}\in T$,算法 k+1步显然正确.



归纳步骤(续)

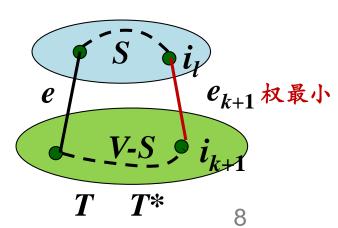
假设 T 不含有 e_{k+1} ,则将 e_{k+1} 加到 T 中形成一条回路. 这条回路有另外一条连接 S 与 V–S中 顶点的边 e,

令
$$T^*=(T-\{e\})\cup\{e_{k+1}\}$$

则 T^* 是 G 的一棵生成
树,包含 $e_1,e_2,...,e_{k+1}$,且

$W(T^*) \le W(T)$

算法到 k+1步仍得到最小生成树.



时间复杂度

算法步骤执行O(n)次

每次执行O(n)时间:

找连接 S与V-S 的最短边

算法时间: $T(n) = O(n^2)$

小结

• Prim算法的设计 贪心策略:连接*S与V-S*的最短边 正确性证明:对步数归纳 伪码

• 时间复杂度: *O*(*n*²)