Dijkstra算法的正确性

归纳证明思路

命题：当算法进行到第 k 步时，对于 S 中每个结点 i，

$$dist\,[i] = short\,[i]$$

归纳基础

$k = 1$, $S = \{s\}$, $dist\,[s] = short\,[s] = 0$.

归纳步骤

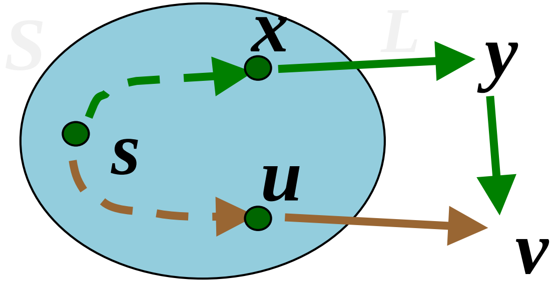
证明：假设命题对 k 为真，则对 $k+1$ 命题也为真.

归纳步骤证明

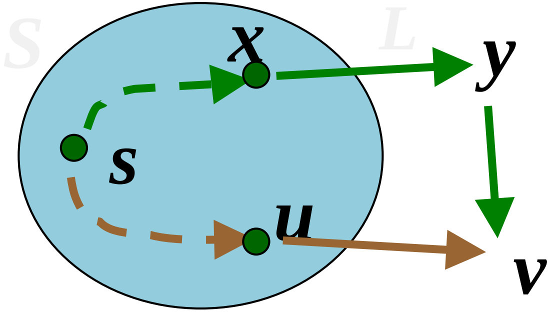
假设命题对*k*为真，考虑 *k*+1步算法选择顶点*v* (边<*u*,*v*>). 需要证明

$\boxed{dist[v]=short[v]}$

若存在另一条 *s*-*v* 路径 *L* (绿色)，最后一次出 *S* 的顶点为 *x*, 经过 *V*–*S* 的第一个顶点 *y*，再由 *y* 经过一段在 *V*– *S* 中的路径到达 *v* .



归纳步骤证明（续）



在 $k+1$ 步算法选择顶点 v，而不是 y，

$$dist\,[v] \le dist\,[y]$$

令 y 到 v 的路径长度为 $d(y,v)$

$$dist\,[y]+d(y,v) \le L$$

于是 $dist\,[v] \le L$，即 $dist[v]= short\,[v]$

时间复杂度

- 时间复杂度：$O(nm)$
  算法进行n-1步
  每步挑选1个具有最小*dist*函数值的结点进入S，需要 $O(m)$时间

- 选用基于堆实现的优先队列的数据结构，可以将算法时间复杂度降到$O(mlogn)$

贪心法小结

- 贪心法适用于组合优化问题.
- 求解过程是多步判断过程，最终的判断序列对应于问题的最优解.
- 判断依据某种“短视的”贪心选择性质，性质的好坏决定了算法的正确性. 贪心性质的选择往往依赖于直觉或者经验.

贪心法小结（续）

- 贪心法正确性证明方法：

  (1) 直接计算优化函数，贪心法的解恰好取得最优值

  (2) 数学归纳法（对算法步数或者问题规模归纳）

  (3) 交换论证

- 证明贪心策略不对：举反例

贪心法小结（续）

- 对于某些不能保证对所有的实例都得到最优解的贪心算法（近似算法），可做参数化分析或者误差分析.
- 贪心法的优势：算法简单，时间和空间复杂性低

贪心法小结（续）

- 几个著名的贪心算法

  最小生成树的**Prim**算法

  最小生成树的**Kruskal**算法

  单源最短路的**Dijkstra**算法