贪婪技术 贪心算法

贪心是一种策略，要证明贪心策略能得到最优解。

分而治之、动态规划（建模递推公式）

贪心，在求解问题的每一步使用贪心策略，注意跟分解子问题区分

找零问题：change-making problem

d1=1, d2=5, d3=10, d4=25, 找48

贪：1 （25）， 2（10），3（1）

6是最少的硬币数吗？

D1=1，d2=10, d3=25, 找30

1（25），5（1）

3（10）

证明：m中面值的硬币，找n元钱，按照这种贪心策略，得到的找零硬币数是最少的

通过一系列的步骤构造问题的解，每一步对目前构造的部分解做一个扩展，直到获得问题的完整解为止。

每一步选择要满足：

1. 可行的（feasible）：满足问题约束
2. 局部最优（locally optimal）：当前步骤中所有可行选择中最佳的局部选择
3. 不可取消（irrevocable）：选择一旦做出，在算法的后面步骤中就无法改变

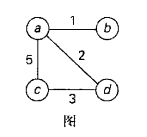
最小生成树 Prim算法，Kruskal算法

加权图最短路径问题 Dijkstra算法

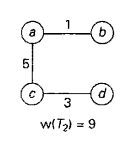
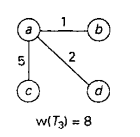
Huffman编码

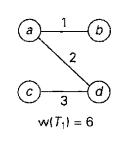
一个连通图的生成树：包含

所有顶点的连通无环子图。



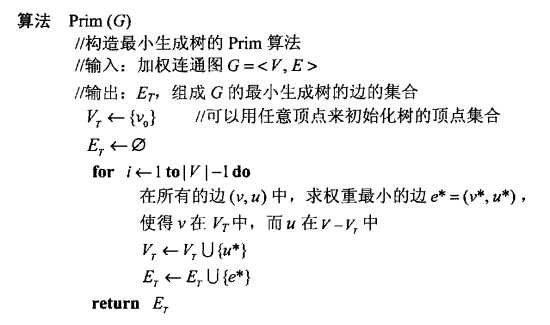
最小生成树：权重和最小

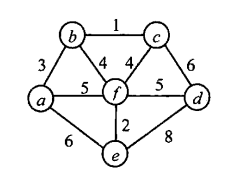
 



贪心：

prim算法





{f,e,b, c,a,d}

{f-e,f-b, b-c,a-b,f-d}

这颗树是最小的嘛？

归纳法

证明：

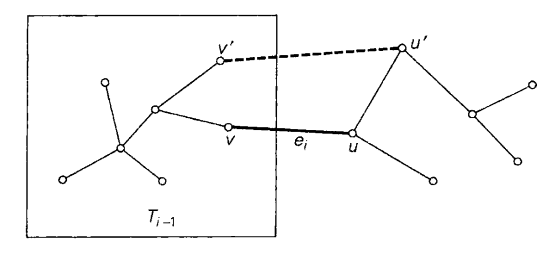
归纳前提：T0包含任意一个顶点，显然T0必定是最小生成树的一部分。

归纳假设：Ti-1是某些最小生成树T的一部分。

根据Prim算法从Ti-1生成Ti，证明Ti也是一颗最小生成树的一部分。如果i从1到n-1都成立，那么Tn-1就是最小生成树。

反正法

假设图中没有一颗最小生成树包含Ti。Ei=（v，u）是从Ti-1中的一个顶点到不在Ti-1中的一个顶点的权重最小的边，Prim算法用它来扩展Ti-1到Ti。因为假设ei不可能属于包括T在内的任何最小生成树。所以如果把ei加到T中，就会构成一条回路。



除了边ei=（v，u），该回路必定包含另一条边（v‘，u’），它把一个顶点v‘（在Ti-1和不在Ti-1）顶点u’连接起来。删除v'u'，得到一颗最小生成树，其权重小于T。

正确性证明

启发式方法

效率（好不好，复杂性分析）