§7.3 贪心算法

一.思想：贪心算法是按照某种标准，先考查最合意的情形，再考查第2合意、第3合意、的情形，依此下去。

例如：假如有4种硬币，面值分别为2角5分，1角，5分和1分。要找给用户6角3分零钱，使得硬币数尽可能少。方案如下：2个2角5分；1个1角；3个1分。每次都选尽可能大额的硬币。

但是贪心算法并不总能够得到最优解。例如：硬币的面值为：1角1分，5分，1分。要找用户1角5分钱，用贪心算法的方案是：1个1角1分；4个1分。但实际最优方案是：3个5分。

二．求最小生成树

1. 问题：在给定的赋权连通无向图中找一棵生成树，使得该树上各边权之和最小。

2. 例子：

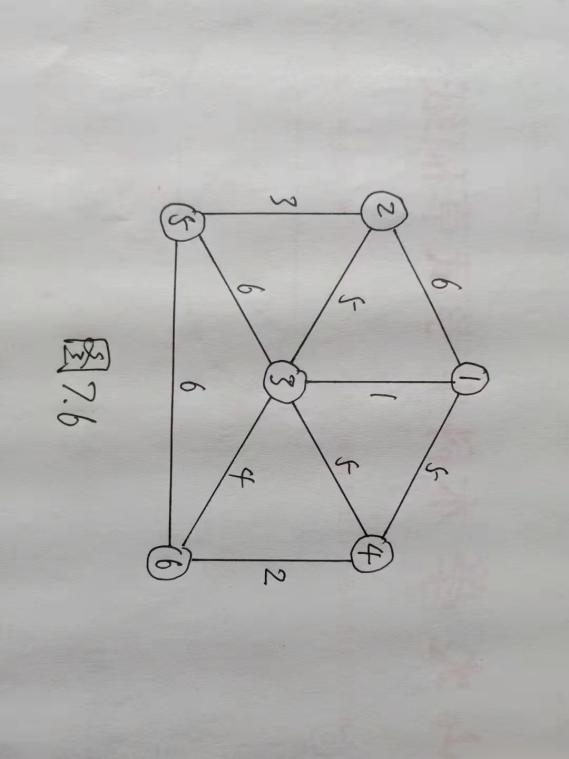


图7.6

3. Kruskal算法：

(1) 将图G的边按权从小到大排序得；

(2) E(T) := ;

(3) FOR i := 1 TO DO

(4) 若与E(T)中的边构成圈，则舍弃;

否则，E(T) := E(T) ;

\*见图7.6

三．活动安排问题：

1. 问题：

设有n个活动的集合，其中，每个活动都要求使用同一资源，例如：演讲会场。而同一时间内只能有一个活动使用这一资源。设每个活动使用该资源的起始时间是，终止时间是，且。若与不相交，则称活动与活动是相容的。也就是说，当或时，活动与相容。

活动安排问题就是要在所给的活动集合中选出最大的相容活动子集合。

2. 算法：

首先将E中所有活动按完成时间从小到大排序。不妨设。如果所给出的活动未按此排序，可用时间先排好序。

PROCEDURE greedy\_Selector (VAR s, f : ARRAY [1n] OF integer;

VAR a : ARRAY [1n] OF boolean);（为 真即被选上）

VAR

n, count, j : integer;

BEGIN

n := |E| ;

a[1] := true ;（一定选第一个活动）

j := 1 ;（j，当前选的最后一个活动）

count := 1 ;

FOR i := 2 TO n DO

IF s[i] f[j] THEN

BEGIN

a[i] := true ;

j := i ;

count := count + 1 ;

END

ELSE a[i] := false;

RETURN(count) ;

END;

3. 例子：（见教材P87, 图4-1）。

4. 算法的正确性证明：

用数学归纳法证明。

设为所给的活动集合。由于中活动按结束时间的非递减顺序排列，故活动1具有最早的完成时间。首先证明活动安排问题有一个最优解以贪心选择开始，即该最优解中包含活动1。设是活动安排问题的一个最优解，且中活动也按结束时间非减序排列，中第一个活动是。若，则就是以贪心选择开始的最优解；若，则设。由于，且中活动是相容的，故中活动也是相容的。也就是说，是以贪心选择活动1开始的最优活动安排方案。

进一步，在做出了贪心选择，即选择了活动1后，原问题简化为对中所有与活动1相容的活动进行活动安排的子问题。也就是说，若是原问题的最优解，则是活动安排问题的最优解。事实上，如果能找到的一个解，它包含比更多的活动，则将活动1加入中，将产生一个解，它包含比更多的活动。这与的最优性矛盾。因此，每一步所做出的贪心选择都将问题简化为一个更小的与原问题具有相同形式的子问题。对贪心选择次数用数学归纳法即知，贪心算法greedy\_Selector最终产生原问题的最优解。

四．贪心算法作为启发技术

在很多问题中，求最优解的算法是指数时间的，我们常用贪心算法作为启发技术，得到一个算法，它在通常情况下运行得很快，并能得到一个“好的解”(不一定最优)。

1. 旅行售货员问题(TSP)

给出n个城市，及城市与城市之间的距离，找出一个旅行方法，访问每个城市恰好一次并回到出发点，使得总的旅途最短。即在赋权无向连通图中找一个最短Hamilton圈。

\*该问题是NP-难题，目前尚无有效算法。

2. 例子：给出六个城市，各点坐标如下：

c (1, 7) d (15, 7)

b (4, 3) e (15, 4)

a (0, 0) f (18, 0)

距离矩阵： 图7.7

3.算法：

(1) 将所有边按权从小到大排序得;

(2) E(C) := ;

(3) FOR i := 1 TO DO

如果(1) 与E(C)中的边不会使某个顶点的度大于等于3；且

(2) 与E(C)中的边不会构成圈，除非是Hamilton圈,

那么E(C) := E(C) ；

否则，舍弃 。

4. 例子：

用2中的例子说明。最短Hamilton圈：acbdefa，圈长=49.8；

而用本算法求出的Hamilton圈为：abcdefa，圈长=50。

§7.4 局部查找算法

一．局部查找算法的思想：

有时按以下策略，可以得到某个问题的最优解：

1. 从任一解开始；
2. 从当前的解变换得到一组解，它们改进当前解。用一个改进的解代替当前解，它成为新的当前解。
3. 反复上述过程，直到变换不产生新的改进解。

\*如果变换后得到的一组解包含所有可能的解，那么，我们最终必然得到最优解。但是这样可能使算法变成指数时间的。

局部查找技术的思想：我们只取一个小的变换后的解集(该变换只用很短时间，例如：或)，这种变换后得到的解“接近”原来的解。该变换称为“局部变换”，这种方法称为“局部查找”。局部查找得到的解不一定是最优解，但常常是“好的解”。

二．用局部查找算法求最小生成树

例1：

图7.8

算法思想：取一条不在树中的边加到树中，得到唯一圈，将该圈中权最大的边删除，得到一棵新树，看新树是否改进了原来的树。

三．局部查找近似算法

例2：TSP问题。 见图7.8

变换方法：

图7.9 图7.10

给出Hamilton圈H，如果存在边AC和BD，则用AC和BD代替H上的边AB和CD，得到一个新的Hamilton圈，看是否改进H。

初始解：（见图7.10）

考虑边对(i)ad, ce; (ii)ac, be，得到局部最优解。该解碰巧是该TSP问题的最优解。

作业15：

1.给定背包问题：给定，要求找出一个元向量, ，使得，而且达到最大。用贪心算法设计一个时间的算法，求解背包问题。

2.设赋权连通图G的边按权从小到大排列是，且任意两条边的权不相等。证明：如果在局部查找算法中，按这个顺序把边加入初始解(一棵生成树)，最后必然得到最优树。