算法设计与分析第四次作业

曲宇勋 201928014628016 人工智能学院

1. 设有 n 个顾客同时等待一项服务。顾客 i 需要的服务时间为 $t_i, 1 \le t_i \le t_n$。应该如何安排 n 个顾客的服务次序才能使总的等待时间达到最小？总的等待时间是各顾客等待服务的时间的总和。试给出你的做法的理由（证明）。

Solution:

假设对顾客的服务时间升序排列，从服务时间小到大第 k 个顾客的序号为 a_k. 那么按照时间从小到大安排，即安排序列为 $\{a_1, a_2, ...a_n\}$ 时，其总的等待时间最小.

总的等待时间为

$$t = \sum_{i=1}^{n}(n-i)t_{a_i}$$

易知总的等待时间最小的一个充分必要条件是,若调换之中任意两者 $a_i, a_j(i < j)$ 的顺序得到序列 $\{a_1, a_2, ...a_{i-1}, a_j, a_{i+1}, ...a_{j-1}, a_i, a_{j+1}, ...a_n\}$. 其等待时间变长，且满足这一条件的序列只有一个.

假设调换之后的等待时间为 t'，由于 $i < j$，所以 $t_{a_i} < t_{a_j}$，那么有

$$t - t' = (n-i)(t_{a_i} - t_{a_j}) + (n-j)(t_{a_j} - t_{a_i}) = (i-j)(t_{a_j} - t_{a_i}) < 0$$

所以 $t < t'$，等待时间会变长. 反之在调换 i, j 的前提下，若 $t < t'$ 则有 $t_{a_i} < t_{a_j}$.

当遍历不同的 $i, j(i < j)$ 时，其需要满足的条件变为对于任意 $i < j$, 需要 $t_{a_i} < t_{a_j}$. 满足这一条件的只有升序排列一种排列.

所以，按升序排列可使得总的等待时间最小.

2. 字符 $a \sim h$ 出现的频率分布恰好是前 8 个 $Fibonacci$ 数，它们的 $Huffman$ 编码是什么？将结果推广到 n 个字符的频率分布恰好是前 n 个 $Fibonacci$ 数的情形。$Fibonacci$ 数的定义为：$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}, n > 1$

Solution:

前 8 个 $Fibonacci$ 数为 $1, 1, 2, 3, 5, 8, 13, 21$，对这组频率生成霍夫曼树可得其对应霍夫曼编码.

表 1: 字符出现频率统计和霍夫曼编码

字符	a	b	c	d	e	f	g	h
频率	1	1	2	3	5	8	13	21
变长码	0000000	0000001	000001	00001	0001	001	01	1

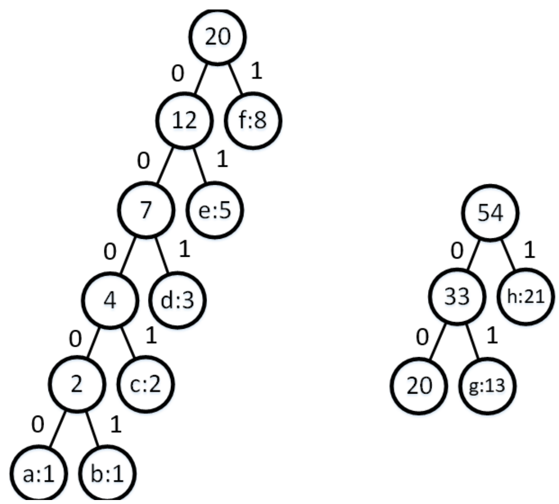

图 1: Huffman 编码树，为节约作图空间右图从根节点出发到 20 以后部分的树结构由左图画出

对于一般情况而言，具有以下对应关系

对于第 1 个字符 a，其编码为 00...0(共 $n-1$ 个 0).

对于第 i 个字符，其编码为 00...01(共 $n-i$ 个 0).

3．设 $p_1, p_2, ..p_n$ 是准备存放到长为 L 的磁带上的 n 个程序，程序 p_i 需要的带长为 a_i。设 $\sum_{i=1}^{n} a_i > L$，要求选取一个能放在带上的程序的最大子集合（即其中含有最多个数的程序）Q。构造 Q 的一种贪心策略是按 a_i 的非降次序将程序计入集合。

1) 证明这一策略总能找到最大子集 Q，使得 $\sum_{p_i \in Q} a_i \leq L$。

2) 设 Q 是使用上述贪心算法得到的子集合，磁带的利用率可以小到何种程度？

3) 试说明 1) 中提到的设计策略不一定得到使 $\sum_{p_i \in Q} a_i \leq L$ 取最大值的子集合。

Solution:

1) 设计的策略为: 将程序 p_i 按照带长从小到大的顺序排列，假设排列顺序为 $B=\{b_1,b_2,...,b_n\}$，取前 k 个程序放入磁带中，其满足

$$\sum_{i=1}^{k} a_{b_i} \le L, \sum_{i=1}^{k+1} a_{b_i} > L,$$

假设有另外一个序列 $C=\{c_1,c_2,...,c_n\}$ 可得到的程序个数比该排列顺序多 k' 个，也即

$$\sum_{i=1}^{k+k'} a_{c_i} \le L, k' > 0$$

那么对该序列的前 $k+k'$ 个元素从小到大排序得到序列 $D=\{d_1,d_2,...,d_{k+k'}\}$，可知 B 是 D 的一个子序列且满足 $a_{d_i} \ge a_{b_i}$.
所以

$$\sum_{i=1}^{k+k'} a_{d_i} \ge \sum_{i=1}^{k+k'} a_{b_i} \ge \sum_{i=1}^{k+1} a_{b_i} > L$$

说明序列 D 也即序列 C 不满足，与假设相矛盾，所以 B 即为最优解.
2) 可以趋近于 0，假设一个具有两个程序的序列，其中 $a_1=l_1, a_2=L-\frac{l_1}{2}$，且认为 l_1 是一个很小的数，那么其满足 $a_1+a_2>L$, 磁盘选择的程序为 p_1，当 l_1 趋近于 0 的时候，磁盘的占用率 $\dfrac{a_1}{L}$ 趋近于 0.
3) 与 2) 中举例相同，令 $l_1=\dfrac{L}{4}$, 那么

$$a_1=\frac{L}{4}, a_2=\frac{7}{8}L$$

贪心算法选择的序列为 p_1，即 $Q=\{p_1\}$. 此时

$$\sum_{p_i\in Q} a_i = \frac{L}{4}$$

但如果选择 p_2 作为选择的序列，$Q'=\{p_2\}$，那么

$$\sum_{p_i\in Q'} a_i = \frac{7L}{8}$$

显然 Q' 要明显优于 Q. 所以贪心策略不一定得到使 $\sum_{p_i\in Q} a_i \le L$ 取最大值的子集合.
6. 说明最优生成树问题具有拟阵结构，并给出赋值函数，解释 $Prim$ 算法和 $Kruskal$ 算法都能求得最优解。

Solution:

令 $S=\{e_1,e_2...e_n\}$ 表示边的集合，$\mathcal{J}$ 表示 S 的部分边构成的所有无圈图构成的集合.

逐一验证拟阵结构的三个条件

(1)S 有限非空

这一点是显然的，S 中包含元素个数为 n，n 为边的条数.

(2) 遗传性质

假设 $B\in\mathcal{J}$, 则表示 B 是由 S 的一个子集 $S_a=\{e_{a_1},e_{a_2}...e_{a_k}\}$ 构成的一个无圈图.

若 $A\subseteq B$，那么作为 B 的子图，A 一定是由 S_a 的子集的边构成的无圈图，也一定是 S 的子集的边构成的无圈图.

所以 $A\in\mathcal{J}$.

(3) 交换性质

假设 $A<B\in\mathcal{J},|A|<|B|$, 有以下两种情况

(1)B 图中的边的顶点集合 D_B 中存在 A 图的边的顶点集合 D_A 中不存在的顶点 d_B, 也即 $d_B\notin D_A,d_B\in D_B$

取 B 图中另外任意一个与 d_B 相连的边 e_B，那么 $e_B\notin A$ 且 $A\cup\{e_B\}$ 一定是无圈的.

(2)B 图中的边的顶点集合 D_B 包含于 A 图的边的顶点集合 D_A, 也即 $D_B\subseteq D_A$.

那么假设 A 图有 k 个相互之间无连接，但内部联通的子图，则每个子图内都是树结构，那么 $|A|=\sum_k(d_k-1)$，其中 d_k 表示第 k 个子图的顶点数.

那么由于 B 中的顶点完全属于 A 中的顶点，假设 B 图中不包含连接某两个子图的边，且无圈结构，那么 B 的边数 $|B|$ 满足 $|B|\leq\sum_k(d_k-1)=|A|$，这与题设相悖，所以 B 图中至少包含一条某两个子图的边 e_B，那么 $e_B\notin A$ 且 $A\cup\{e_B\}$ 一定是无圈的.

所以存在 $e_B\in B,e_B\notin A,A\cup\{e_B\}\in\mathcal{J}$.

以上三条满足，所以最小生成树问题符合拟阵结构.

每一条边的赋权即为每条边的边权.

对于 $Kruskal$ 方法而言，相当于对于之前定义的拟阵结构 $(S,\mathcal{J})$，每次取出权值最大的元素后求取其子问题，其符合拟阵结构的贪心算法解法，一定具有最优解.

对于 $Prim$ 方法而言，令 $S=\{e_1,e_2...e_n\}$ 表示边的集合，$\mathcal{J}$ 表示 S 的部

分边构成的所有树图构成的集合. 也可以证明出其具有拟阵结构，所以采用贪心算法也一定具有最优解.