

计软实验二：贪心算法

一、实验目的

通过编程熟悉并掌握贪心算法，了解其在背包问题以及哈夫曼编码问题上的应用。

二、实验内容

1.背包问题

背包问题的定义为：有 n 个物品，它们有各自的重量和价值，现有给定容量的背包，如何让背包里装入的物品具有最大的价值总和？根据物体可否分割性质，背包问题可分为两类，一类是分数背包问题，每个物体都是可分割的；另一类是 01 背包问题，即物体不可分割。

数学定义为：设结果集为 $X = (x_0, x_1, \dots, x_{n-1})$ ，约束条件为 $\sum_{i=0}^{n-1} w_i x_i \leq M$ ，目标函数为 $\max \sum_{i=0}^{n-1} v_i x_i$ ，其中 w_i 为第 i 个物体的重量， v_i 为第 i 个物体的价值，对于分数背包问题 $x_i \in [0,1]$ ，对于 01 背包问题 $x_i \in \{0,1\}$ 。

贪心算法的思想是将价值重量比大的物品放入背包，对于可分割的物品即对物品进行价值重量比排序，按顺序放入物品直至背包装满。而对于不可分割的物品则直接按照价值排序，将价值高的物品优先放入背包。

（1.1）分数背包问题：

已知有 7 个物品，每个物品可以分割成任意大小，现要将物品装进容量为 150 的背包，要求尽可能让装入背包中的物品总价值最大，但不能超过总容量。物品信息如下：

物品	1	2	3	4	5	6	7
重量	35	30	60	50	40	10	25
价值	10	40	30	50	35	40	30

要求：用贪心算法进行求解，输出为装进背包每种物品的重量以及总价值。

（1.2）01 背包问题：

已知有 7 个物品，每个物品不可分割（要么选要么不选），现要将物品装进容量为 150 的背包，要求尽可能让装入背包中的物品总价值最大，但不能超过总容量。物品信息如下：

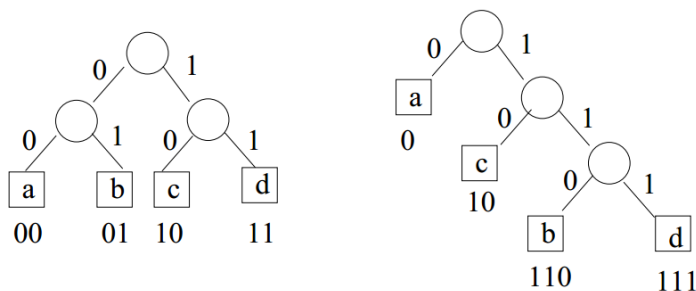
物品	1	2	3	4	5	6	7
重量	35	30	60	50	40	10	25
价值	10	40	30	50	35	40	30

要求：用贪心算法进行求解，输出为装进背包每种物品的编号以及总价值。

2.哈夫曼编码问题

哈夫曼编码是可变字长编码的一种，该方法完全依据字符出现概率来构造异字头的平均长度最短的码字。问题定义为：给定一个字母表 $A = (a_1, a_2, \dots, a_n)$ 并且已知每个字母的频率寻找一个二进制前缀码 C ，使得 C 对 A 的编码长度最短。即： $B(C) = \min(\sum_{i=1}^n f(a_i)L(c(a_i)))$ ，其中 $f(a_i)$ 为 a_i 出现的频率， $c(a_i)$ 为对 a_i 的编码， $L(c(a_i))$ 为对应的编码长度。

哈夫曼编码可以用哈夫曼树来表示。哈夫曼树从根节点到叶节点的路径构成的二进制字符串即是该叶节点对应的码字，如下图，左图叶节点 a 的路径为 00 即为码字，叶节点 b 对应 01 ，以此类推。



结合哈夫曼树，问题定义变为：对于树 T 的加权路径长度 $B(T)$ 为： $\sum_{i=1}^n f(a_i)d(a_i) = \sum_{i=1}^n f(a_i)L(c(a_i))$ ，给定权重 $f(a_i)$ ，找到树 T 使得它的加权路径长度最小。

贪心算法可以用于构造哈夫曼编码，方法为自底向上构造哈夫曼树。首先从字母表 A 中选择频率最小的两个字符 x, y ，然后创建一个子树，保护 x, y 两个叶节点，并将子树的根标记为 z 。 z 的频率 $f(z) = f(x) + f(y)$ ， $A' = A \cup \{z\} - \{x, y\}$ ， $|A'| = |A| - 1$ 。然后用新的字母表 A' 重复以上步骤，直到只剩下一个符号的字母为止，最终得到的树即哈夫曼编码结果。

字母表如下：

字符	a	b	c	d	e	f	g
频率	0.25	0.1	0.12	0.2	0.15	0.07	0.11

要求：用贪心算法求每个字符对应的哈夫曼编码

附加题：动态规划

实验内容

贪心算法只能在分数背包问题上得到最优解，而无法保证 01 背包问题的最优解。动态规划可以用于解决 01 背包问题并得到最优解。

最优性原理是动态规划的基础，最优性原理是指“多阶段决策过程的最优决策序列具有这样的性质：不论初始状态和初始决策如何，对于前面决策所造成的某一状态而言，其后各阶段的决策序列必须构成最优策略”。设 $m[i][j]$ 表示对于前 i 个物体，背包容量为 j 时所能装下物品的最大价值。则 $m[i][j]$ 的计算方法为

$$m[i][j] = \begin{cases} \max(m[i-1][j-w[i]] + v[i], m[i-1][j]) & j \geq w[i] \\ m[i-1][j] & j < w[i] \end{cases}$$

《计算机软件基础》实验指导书

计算得到 $m[n][M]$ 即为背包的最大价值。

要求：用动态规划求解 01 背包问题，输出为装进背包每种物品的编号以及总价值，并与贪心算法的结果进行比较。