第五章 贪心法作业

第一题

- （0/1背包问题）如果将背包问题修改成

  极大化 $\sum_{1}^{n} p_i x_i$

  约束条件 $\sum_{1}^{n} w_i x_i \le M$

  其中，x_i=0或1　$1 \le i \le n$

- 这种背包问题称为0/1背包问题。它要求物品或者整件装入背包或者整件不装入。求解此问题的一种贪心策略是：按p_i/w_i的非增次序考虑这些物品，只要正被考虑的物品能装的进就将其装入背包。证明这种策略不一定能得到最优解。

- 证明:
  - 当按照p_i/w_i的非增次序考虑物品存放背包时，如果所装入的物品能装满背包时，显然为最优解，否则未必是最优解.
  - 可举例如下:

    设n=3，M=6，$(p_1,p_2,p_3)=(3,4,8)$，$(w_1,w_2,w_3)=(1,2,5)$

    按照p_i/w_i 的非增序得到

    （$p_1/w_1,p_2/w_2,p_3/w_3$） =(3,2,1.6),

    则其解为（1,1,0），而事实上最优解是(1,0,1) 。

    命题得证。

第二题

- 已知定理5.3面向时间片相同的作业调度问题：
  - 设J是k个作业的集合, $\sigma=i_1,i_2,\ldots,i_k$是J中作业的一种排列, 它使得$d_{i1}\le d_{i2}\le\ldots\le d_{ik}$。J是一个可行解, 当且仅当J中的作业可以按照$\sigma$的次序而又不违反任何一个期限的情况来处理。
- 证明即使作业有不同时间片，该定理亦真。规定作业i的效益pi>0, 要用的处理时间ti>0，限期di≥ti。

- 证明思想：
  - ←
  - → 位置a,b的作业交换顺序
    - 作业ra和rb仍然可以完成任务
    - 作业ra和rb之间的作业也能够完成任务

证明：

- 显然即使作业的处理时间各不相同（$t_i>0$，$d_i \geq t_i$），如果J中的作业可以按照σ的次序排列而又不超出任何一个期限的限定，即对σ次序中的任一个作业k，满足$d_k \geq \sum_{j=1}^{k} t_j$，则J就是一个可行解。
- 下面证明如果J是可行解，$\sigma = i_1 i_2 \ldots i_k$使得J中的作业可以按照$d_{i1} \leq d_{i2} \leq \ldots \leq d_{in}$排列而又不超出任何一个作业的期限。

- 若J是可行解，则必存在一种排列$\sigma'=r_1r_2\ldots r_n$，使得对任意的r_k，都有 $d_{r_k} \geq \sum_{j=1}^{k} t_j$ 。
- 设σ是按照$d_{i1} \le d_{i2} \le \ldots \le d_{in}$排列的作业序列。若$\sigma'=\sigma$，则σ中每个作业都不会超出其期限。命题得证.
- 假设$\sigma' \neq \sigma$，那么令a是使$r_a \neq i_a$的最小下标，设$r_b=i_a$，显然b>a. 在σ'中将r_a与r_b相交换，因为$d_{rb} \le d_{ra}$，交换后r_a和r_b可以在各自的期限前完成作业。

- 下面证明r_a和r_b之间的作业也能按期完成。因为$d_{rb} \le d_{ra}$，而显然二者之间的所有作业r_t，都有$d_{rb} \le d_{rt}$，又由于σ'是可行解，所以$\sum_{j=1}^{b} t_j \le d_{r_b} \le d_{r_t}$，$r_a$和$r_b$交换后$r_a$和$r_b$之间的作业也能按期完成。
- 作业r_a和r_b交换后，所有作业可依照新产生的排列$\sigma'' = s_1 s_2 \ldots s_n$的次序处理而不违反任何一个期限，连续使用这种方法，σ'就可转换成σ且不违反任何一个期限，定理得证。

第三题

- 对于带有期限的作业调度问题：
  - n=7
  - (p1,…,p7)=(3,5,20,18,1,6,30)
  - (d1,…,d7)=(1,3,4,3,2,1,2)
- 基于直接插入排序的贪心算法(算法5.4)所生成的解是什么？请给出J的变化过程。

- n=7
- (p1,…,p7)=(3,5,20,18,1,6,30)
- (d1,…,d7)=(1,3,4,3,2,1,2)
- 解：按照P从大到小排序：(p_7, p_3, p_4, p_6, p_2, p_1, p_5)=(30,20,18,6,5,3,1)，对应期限为(2,4,3,1,3,1,2)

一维数组J	1	2	3	4	5	6	7
①. 计入作业7	7						
②. 计入作业3	7	3					
③. 计入作业4	7	4	3				
④. 计入作业6	6	7	4	3			

其他舍弃

第四题

- 对于带有期限的作业调度问题：
  - 作业数n＝7
  - $(p_1,p_2,\ldots\ldots, \ p_7) = (40, 12, 30, 20, 7, 15, 10)$
  - $(d_1,d_2,\ldots\ldots, \ d_7) = (2, 4, 4, 3, 2, 1, 6)$
- 利用基于集合树的贪心算法(算法5.5)，求解上述作业调度问题的最优解。（要求按步骤运行并给出集合树的变化情况、最优解及效益值）。

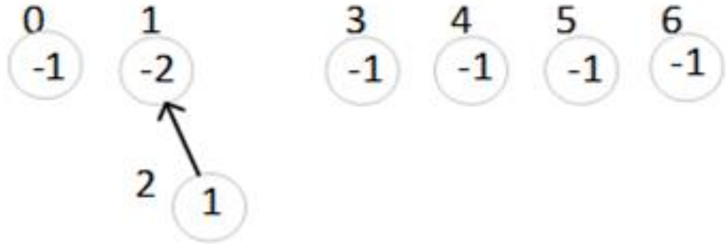
- 对于带有期限的作业调度问题：
  - 作业数n＝7
  - （$p_1,p_2,\ldots\ldots, p_7$）＝（40, 12, 30, 20, 7, 15, 10）
  - （$d_1,d_2,\ldots\ldots, d_7$）＝（2, 4, 4, 3, 2, 1, 6）
- 解
  - 按P值排序
  - （$p_1,p_3, p_4,p_6, p_2,p_7,p_5$）＝（40, 30, 20, 15, 12, 10 , 7）
  - 对应的期限为（2, 4, 3, 1, 4, 6, 2）
  - $b = \min\{ n, \max\{d(i)\} \} = \min\{7, 6 \}= 6$

$(p_1, p_3, p_4, p_6, p_2, p_7, p_5) = (40, 30, 20, 15, 12, 10, 7)$，对应的期限为（2, 4, 3, 1, 4, 6, 2）

	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
	0	1	2	3	4	5	6

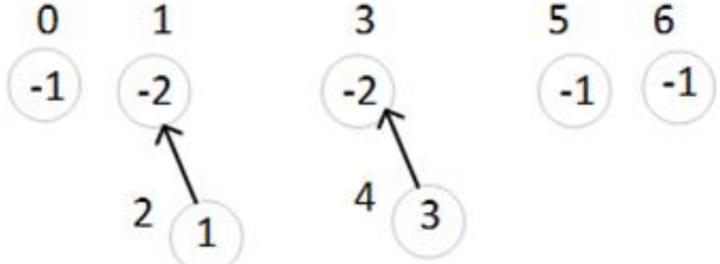
J
{ }

作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
1	0	1	1	3	4	5	6



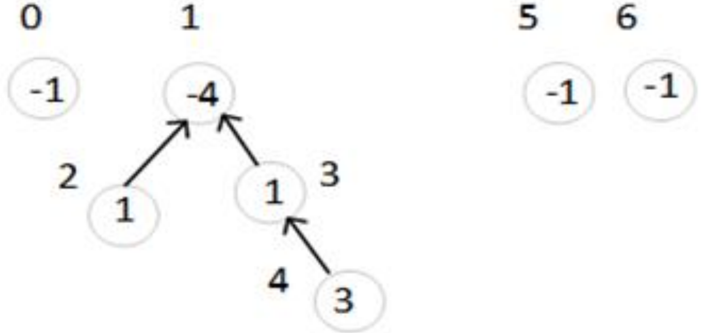
J
{ 1}

作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
3	0	1	1	3	3	5	6



J
{ 1,3}

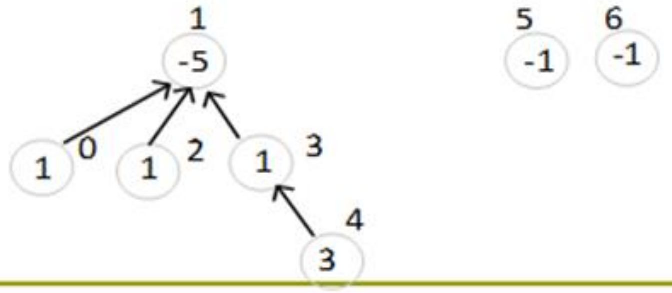
作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
4	0	1	1	1	3	5	6



J
{ 1,3,4}

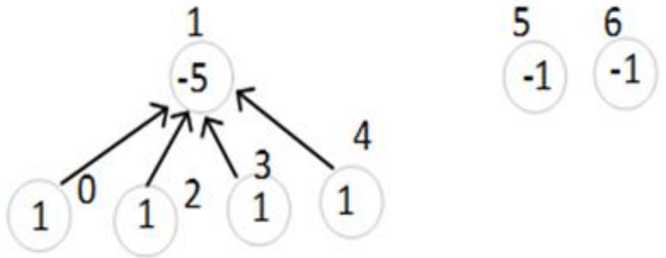
$(p_1,p_3, p_4,p_6, p_2,p_7,p_5) = (40, 30, 20, 15, 12, 10, 7)$，对应的期限为（2, 4, 3, 1, 4, 6, 2）

作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
6	0	0	1	1	3	5	6



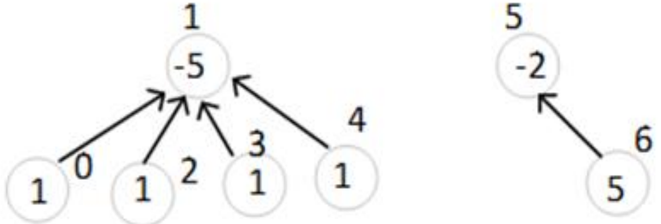
J
{ 1,3,4,6}

作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
2(舍弃)	0	0	1	1	1	5	6



J
{ 1,3,4,6}

作业	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)
7	0	0	1	1	1	5	5



J
{ 1,3,4,6,7}

作业
5(舍弃)

选中的作业为1,3,4,6,7。效益值为115

第五题

- 假定要将长为$l_1,l_2,\ldots,l_n$的n个程序存入一盘磁带，程序i被检索的频率是f_i。如果程序按$i_1,i_2,\ldots,i_n$的次序存放，则期望检索时间（ERT）是:

$$[\sum_{j}(f_{ij}\sum_{k=1}^{j}l_{ik})]/\sum f_i$$

- ① 证明按l_i的非降次序存放程序不一定得到最小的ERT。
- ② 证明按f_i的非增次序存放程序不一定得到最小的ERT。
- ③ 证明按f_i/l_i的非增次序来存放程序时ERT取最小值。

① 证明按li的非降次序存放程序不一定得到最小的ERT。

- 举例如下:
  - (l1,l2)=(10,12)
  - (f1,f2)=(0.4,0.6)
- ERT(I)=10*0.4+(10+12)*0.6=17.2
- ERT(I')=12*0.6+(10+12)*0.4=16

② 证明按fi的非增次序存放程序不一定得到最小的ERT。

- 举例如下:
  - (l1,l2)=(2,1)
  - (f1,f2)=(0.6,0.4)
- ERT(I)=2*0.6+(2+1)*0.4=2.4
- ERT(I’)=1*0.4+(1+2)*0.6=2.2

③ 证明按fi/li的非增次序来存放程序时ERT取最小值。

- 假设i1,i2,…,in按照fi/li的非增次序存放，即$f_{i1}/l_{i1} \ge f_{i2}/l_{i2} \ge \ldots \ge f_{in}/l_{in}$，则得到 $ERT=[f_{i1}l_{i1}+f_{i2}(l_{i1}+l_{i2})+\ldots+f_{ik}(l_{i1}+l_{i2}+\ldots+l_{ik})+\ldots+f_{in}(l_{i1}+l_{i2}+\ldots+l_{in})]/(f_{i1}+..+f_{in})$
- 假设该问题的一个最优解是按照$j_1,j_2,\ldots,j_n$的顺序存放，并且其期望检索时间是ERT′，我们只需证明ERT≤ERT′，即可证明按照fi/li的非增次序存放得到的是最优解。

- 从前向后考察最优解序列：$j_1,j_2,\ldots,j_n$，若与$i_1,i_2,\ldots,i_n$相同，命题得证。
- 否则，不妨设程序j_k是第一个与其相邻的程序j_{k+1}存在关系$f_{jk}/l_{jk}<f_{jk+1}/l_{jk+1}$，交换程序jk和程序jk+1，得到的期望检索时间记为ERT″。
- $ERT' - ERT'' = (f_{jk+1}l_{jk} - f_{jk}l_{jk+1})/(f_{i1}+..+f_{in})>0$，既有$ERT'' \le ERT'$，显然ERT″也是最优解。
- 最优解中所有这样类似于反序对的程序互换位置，每次得到的解不比原来的最优解差，所以最终变换后得到的解也是最优解，而最终的解恰是程序按f_i/l_i的非增次序来存放得到的顺序。
- 命题得证。

第六题

- 有编号为1,2,…,n的集装箱准备装上轮船，其中集装箱i的重量是wi，i=1,2,…,n. 已知轮船最大承重量是C，轮船对集装箱没有体积限制。如何选择装载，可使船上集装箱数量最多。
- ①贪心方法可以求得装载问题的最优解吗？若可行，请给出最优量度标准，并证明你的结论。
- ②请给出贪心算法描述，并分析算法的时间复杂度。

问题分析

- 基于目标函数设计“轻者先装”的贪心策略。
- 证明方法1：仿照背包问题
- 证明方法2：仿照作业调度问题

证明方法1：仿照背包问题

- 装载问题形式化表示如下
  - 问题的解表示为X=($x_1, x_2, \ldots, x_n$)，其中x_i =0或1，要求满足 $\sum_{i=1}^{n} w_i x_i \le C$ 的约束条件下，使得可装载的集装箱总数量

$$\sum_{i=1}^{n} x_i \text{取最大值}$$

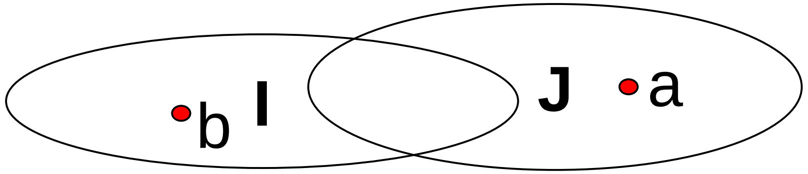
- 提示
  - 按照集装箱wi从小到大顺序为集装箱编号，设X=($x_1, x_2, \ldots, x_n$)是贪心解，x_i =0或1。由贪心算法可知X取值如下所示，j是第一个取值为0的位置。

$$\begin{array}{cccccccc} x_1 & \cdots\cdots & x_{j-1} & \boxed{x_j} & x_{j+1} & \cdots\cdots & x_n \\ 1, & \ldots\ldots, & 1, & x_j, & 0, & \ldots\ldots, & 0 \end{array}$$

- 设最优解 $Y=(y_1,\ldots,y_n)$，比较两者取值，设第一个不同值的位置是$k<j$，则$y_k=0$。修改$y_k=1$，同时在$n\geq m\geq j$范围内寻找值为1的任意y_m，令$y_m=0$，即放弃m选择k。这样构造一个新的解Z。重复该构造过程。证明细节不再赘述，请按下述提示完善补充。
  - $k<j$一定成立么?
  - 一定能找到y_m么?
  - Z会满足约束条件C么?
  - 重复构造直到$y_{j-1}=1$时，最优解在$n\geq m\geq j$范围内是否还存在$y_m=1$？
- 算法设计仿照背包问题和分析时间复杂度。

证明方法2：仿照作业调度问题

- 提示
  - 按照集装箱wi从小到大顺序为集装箱编号，设J是贪心解，记录选中的集装箱编号，I是问题的最优解。不失一般性J和I如下关系：



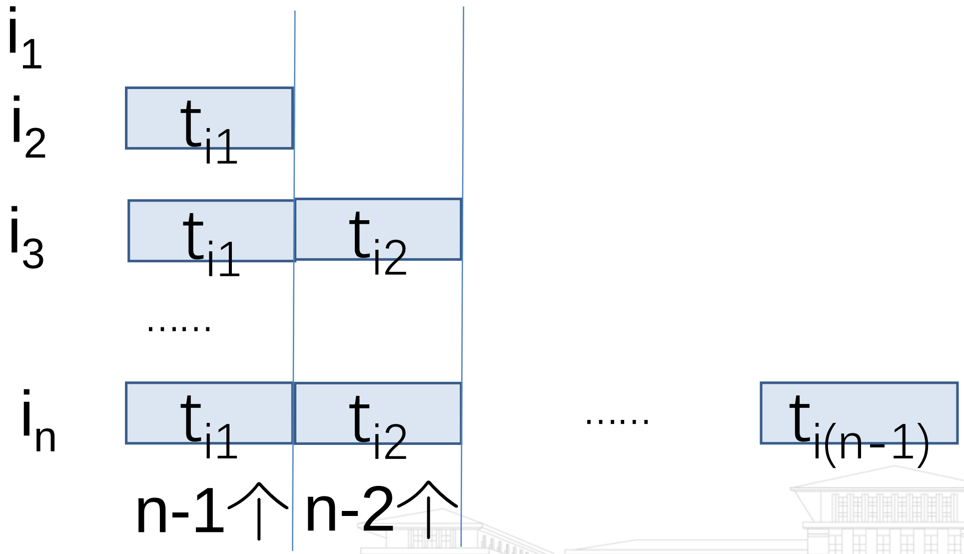
  - a是使得a∈J且a∉I的最小编号的集装箱，b是b∈I且b∉J的最大编号的集装箱，用 a去替换b，获得一个新的解I’。重复该构造过程。证明细节不再赘述，请按下述提示完善补充。
  - a<b一定成立么？即$W_a<W_b$?
  - 一定能找到b么?
  - I’会满足约束条件C么?
  - 当I被J的所有元素都替换完毕后，是否还存在多余元素b？
- 算法设计略。

第七题

- 问题描述：设有 n 个顾客同时等待一项服务，顾客需要的服务时间为 t_i, i=1,2,...,n. 从时刻0开始计时.若在时刻 t 开始对顾客 i 服务，那么 i 的等待时间就是 t. 应该怎样安排 n 个顾客的服务次序，使得总的等待时间（每个顾客等待时间的总和）最少？
  - ①使用贪心法求解这个问题时的贪心选择策略是什么?请证明该策略是最优的。。
  - ②写出贪心法的算法描述，并分析算法的时间复杂度。

(1) 使用贪心法求解这个问题时的贪心选择策略是什么?

- 确定目标函数：若调度 f 的顺序是$i_1,i_2,\ldots,i_n$，那么i_k 的等待时间是$\sum_{j=1}^{k-1} t_{ij}$，总的等待时间就是time(f)=$\sum_{i=1}^{n}(n-i)t_{ij}$,假设调度 f* 使得$t_1\le t_2\le \ldots\le t_n$，按照1,2,…,n 的顺序安排服务，则f*的总等待时间是：time(f*) =$\sum_{i=1}^{n}(n-i)t_i$,



- 确定最优度量标准。即贪心策略应该根据什么进行排序呢？
  - 最优度量标准就是服务时间长短
  - 根据目标函数，贪心策略对服务时间较短的优先安排
  - 服务时间从短到长对顾客进行排序，即f*是最优量度标准
- 如何证明上述贪心策略是正确的呢？即如何证明 f* 排序可以得到最优解呢？
  - 证明思路：假设在最优解g中，顾客i+1在顾客i之后立刻得到服务，$t_{i+1}<t_i$。交换i和i+1得到新的服务顺序g*，那么 time(g) - time(g*)=$t_i – t_{i+1} \geq 0$，总等待时间将不会增加，因此g*也是最优解，不断交换反序对，至多经过 n(n-1)/2 次这样的交换，就可以将g转化成算法的解 f*，从而证明 f* 是最优解。

t_1	......	t_i	t_{i+1}	......	$t_{i(n-1)}$
		n-i个	n-i-1个		

(2)简单写出贪心法的算法描述。

- 算法思想
  - 按照服务时间从小到大预排序
  - 排序后的数组是输入参数
  - 按照$\sum_{i=1}^{n}(n-i)t_i$求解，获得的结果就是问题的最优解。
- 算法复杂度
  - 考虑预排序：时间复杂度为 O(nlogn)
  - 按公式求解：时间复杂度为 O(n)

本 章 结 束