第四章练习参考答案

习题四

1 解：设调度 f 为: $i_1,i_2,\ldots,i_n$，作业 i_k 需等待 $\sum_{j=1}^{k-1} t_{i_j}$ ，即作业 1 等待 0，

作业 2 等待 i_1，作业 3 等待 $i_1+i_2\ldots$.，总的等待时间为：$T(f)=\sum_{j=1}^{n}(n-i)t_{i_j}$ 。使用贪心策略：服务时间短的作业先安排。算法描述略。

正确性：交换论证。不妨假设 $t_1 \leqslant t_2 \leqslant t_2 \leqslant \cdots \leqslant t_n$，算法的调度 f 结果为 1,2,. . , n。设它不是最优的，存在最优调度 f*，设其最早第 k 项作业 i_k 与 f 不同，即 f*:1,2,. . k-1, i_k, $i_{k+1}\cdots i_n$，必有 $t_{ik}>=t_k$。 今将 f* 中作业 K 与作业 i_k 置换，得到调度 f**：$1,2\ldots k,i_{k+1},\ldots i_k\ldots i_n$。其中 i_k 位置为 j，则 j>k, $t_{ik}>=t_k$。则：

$T(f^*)-T(f^{**})=(n-k)t_{ik}+(n-j)t_k-(n-k)t_k-(n-j)t_{ik}=(j-k)t_{ik}+(k-j)t_k=(j-k)(t_{ik}-t_k)>=0$

说明 f**也是最优调度，但它与 f 不同的次序项后移了一位。重复此过程最多 n 步，可得 f 最优。

2.解：a-h 的频率为：1,1,2,3,5,8,13,21,按 huffman 规则，编码为

0000000,0000001,000001,00001,0001,001,01,1

设前 k 个字符符合：$f_0+f_1+..+f_{k-1}<f_{k+1}$ 成立(1<k<8 已成立)，则

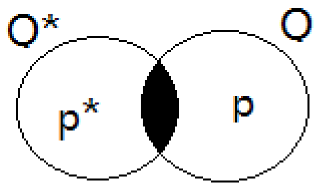
$F_0+f_1+\ldots+f_{k-1}+f_k<f_{k+1}+f_k=f_{k+2}$,定义字符 $z_k= f_0+f_1+..+f_{k-1}$，则对任意 k，$z_k<f_{k+1}$

经 huffman 编码 k-1 步后，得 $z_k,f_k,f_{k+1},\ldots.$,据上，$z_k<f_{k+1}$,而显然 $f_k<f_{k+1}$

Z_k,f_k 是最小频率的两个数，合并，依次类推，每次都是新字符与下一个 Fibonacci 数合并，所以，f0-f_n 的编码为 000…0,000…1,..001,01,1。

其中 f_k 的编码为 000..01，共 n-k 个 0（k>0），f0:n 个 0。

3 解：



(1)反正法。设算法得到的程序集合为 Q 不是最优的，Q*是一个最优集合，则 Q 与 Q*无包含关系。Q 不能包含 Q* 是显然的，如果 Q*包含 Q ，那么对 p∈Q*\Q，如果 p 不小于 Q 的成员，算法将会在接下来的判断中将其选入(不超 L)，否则，它先被检查，早应该是 Q 的成员。

取 Q*使得|Q∩Q*|最大。令 p 是 Q\Q*中最小的，则 Q 中比 p 更小的程序一定在 Q*中(阴影部分)。则任给 p*∈Q*\Q，必有 p*>=P。否则，若 p*<P，p *在 P 前被算法检查，此时磁带装入的程序均在 Q∩Q*即 Q* 中，P*不被选入，说明此时磁带已装不下 p* ，但 P*在 Q*里，说明 P*在 Q∩Q*之外可以装入，矛盾。

将 Q*中的 P*换成 P，显然可以装入，得到 Q**，也是最优的，但|Q**∩Q|>| Q*∩Q|，与 Q*的选取矛盾。得证。

(2)磁带利用率为该是集合 Q 中所有程序的长度之和/L，可以为 0，如果一个也装入不了。

(3)反例，l=8,n=4,a1=1,a2=2,a3=3,a4=4,算法装入 p1,p2,p3,带长浪费 2，而装入 p1,p3,p4，带长浪费 0。

4.见课件。

二、

1.解：使用贪心法：令 $a_1,a_2,\ldots$表示基站的位置。

贪心策略：首先令 $a_1=d_1+4$，对 $d_2,d_3,\ldots d_n$ 依次检查，找到下一个不能被该基站覆盖的房子。如果 $d_k<=a_1+4$ 但 $d_{k+1}>a_1+4$,那么第 k+1 个房子不能被基站覆盖，于是取 $a_2=d_{k+1}+4$ 作为下一个基站的位置，照此下去，直到检查完 d_n 为止。

算法伪码：Location()

输入：距离 $d_1,d_2,\ldots,d_n$ 的数组 d[1..n]:满足 d[1]<d[2]<…<d[n]

输出：基站位置的数组 a[..]

```
a[1]:=d[1]+4
K:=1
for j:=2 to  n
   If d[j]>a[k]+4
   Then k:=k+1
         a[k]:=d[j]+4
Return a
```

算法的正确性证明使用归纳法：对任何正正整数 k，存在最优解包含算法前 k 步选择的基站位置。

证明：k=1，存在最优解包含 a[1]。若不然，有最优解 OPT，其第一个基站位置是 b[1]，b[1]≠a[1]。那么 $d_1-4\leqslant b[1]<d_1+4=a[1]$。B[1]覆盖的是距离在$[d_1,b[1]+4]$之间的房子。A[1]覆盖的是距离$[d_1,a[1]+4]$的房子，因为 b[1]<a[1],b[1]覆盖的房子都在 a[1]覆盖的区域内，用 a[1]替换 b[1]，得到的仍旧是最优解。

假设对于 k 存在最优解 A 包含算法前 k 步的选择的基站，即

$A=\{a[1],a[2]\cdots a[k]\}\cup B$，其中 $a[1],a[2]\cdots a[k]$ 覆盖了距离 $d_1, d_2, \cdots, d_j$ 的房子，那么 B 是关于 $L=\{d_{j+1}, d_{j+2}, \cdots, d_n\}$ 的最优解。否则，存在关于 L 的最优解 B^*，那么用 B^* 替换 B 得得到 A^*，且 $|A^*|<|A|$，与 A 是最优解矛盾。根据归纳假设，L 有一个最优解 $B'=\{a[k+1],\cdots,\}$，$|B'|=|B|$。于是，$A'=\{a[1],a[2],\cdots,a[k]\}\cup B'=$

$\{a[1],a[2],\cdots,a[k+1]\cdots\}$ 也是最优解，且 $|A|=|A'|$，从而证明了结论对 k+1 也真。证毕。

算法复杂度 $T(n)=O(n)$。

2.解：使用贪心法。

贪心策略：把进程按截止时间排序。取第一个进程的截止时间作为第一个测试点，然后顺序检查后续能够被这个测试点检测的进程(这些进程的开始时间小于测试点)，直到找到下一个不能够被这个测试点测试到的进程为止。取这个进程的截止时间作为下一个测试点，……，知道检查完所有的进程为止。算法描述：

Test()

输入：s[1..n]，d[1..n]；输出：t[]，顺序选定的测试点构成的数组

```
  将进程按 d[i]递增顺序排序，使 d[1]≤d[2]≤…≤d[n]
  i:=1,t[i]=d[i]
  j:=2
  do {
  while j<n and s[j]≤t[i] do {
     j:=j+1
```

```
  }
  if j>n then return t
  else
      i:=i+1
      t[i]:=d[j]
      j:=j+1
  end if
until j>n
```

用归纳法证明其正确性:任给 k，存在最优解包含算法前 k 步选择的结果。

证明：k=1，设 s={$t[i_1]$,$t[i_2]$，…}是最优解，则一定 $t[i_1]<t[1]$，否则 $t[i_1]$不能测试 d[1]的进程。设 p_u是在时刻 $t[i_1]$被检测到的任意进程，则 $s[u]\leqslant t[i_1]\leqslant d[u]$，从而 $s[u] \leqslant t[i_1]<t[1]=d[1] \leqslant d[u]$，因此，pu 也可以在 t[1]时刻被测试，在 s 中将 $t[i_1]$替换为 t[1]也是最优解。

假设结论对 k 成立，则存在最优解 $T=\{t[1],t[2],\cdots,t[k]\}\cup T'$。设算法前 k 步选择的测试点不能测到的进程构成集合 $Q\subseteq P$，P 是全部进程集合，那么 T'是问题 Q 的最优解。根据归纳假设，存在 Q 的最优解 T^*包含测试点 t[k+1]，即 $T^*=\{t[k+1]\}\cup T''$，因此，$\{t[1],t[2],\cdots,t[k]\}\cup T^*=\{t[1],t[2],\cdots,t[k],t[k+1]\}\cup T''$也是原问题的最优解，得证。

算法最坏时间复杂度:排序 O(nlogn),检查 O(n),所以 T(n)=O(nlogn)。

3．解：使用贪心法。贪心规则：优先安排前D个罚款最多的作业。

正确性证明：交换论证。设算法选择的作业调度 f 的安排次序是 $\langle i_1, i_2, \cdots, i_n \rangle$，那么罚款为 $F(f)=\sum_{k=D+1}^{n} m(i_k)$，任给 $k<=D<j$，$m(i_k)>=m(i_j)$。

显然最优调度没有空闲时间，不妨设作业是连续安排的。每项作业的加工时间都是 1，在 D 之前可完成 D 项作业，在 D 之后安排的 n-D 项作业 $i_{D+1}, i_{D+2}, \cdots, i_n$ 是被罚款的作业。

设算法得到的安排不是最优的，则存在一个最优调度 f^*，它的前 D 个作业包含了至少 1 个作业 i_j，j>D，从而至少有一个作业 i_k，k<=D 被安排在了 D 之后。交换 i_k, i_j 得到新的调度 f^{**}，则 $F(f^*)-F(f^{**})=m(i_k)-m(i_j)\geqslant 0$，说明 f**也是最优调度，但 f**与 f 的前 D 个相同作业多了 1 个，依次进行，可得最优作业与 f 相同，得证。

算法描述：

算法 A:按 m[i]非升排序，依次选择作业即可。但 T(n)=O(nlogn).

算法 B:m^*:=select(m[],n-D)

Partition(m[],A,B,m^*)

{ $i_1, i_2, \cdots, i_D$ }：=B，{$i_{D+1}, i_{D+2}, \cdots, i_n$}:=A+{$m^*$}

算法复杂度 T(n)=O(n)。

4.解：零钱系统币值为 25,10,1 元，找 30 元。

按贪心算法 30=25+1X5 共 6 枚。但 30=10*3 只要 3 枚。