Question 1. *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 $t_j, 1 \le i \le n$。应该如何安排 n 个顾客的服务次序才能使得总的等待时间最小？总的等待时间是各顾客等待服务时间的总和。试给出你的证明。

My Answer 1. 由题：

设 $J = \{x_1, ..., x_n\}, i = 1, ..., n$ 为第一个顾客、第二个顾客...、第 n 个顾客的编号，$T(J) = \sum_{i=1}^{n-1} t_{x_i}(n-i)$ 即总的等待时间

采用贪心策略：先服务服务时间长的顾客：也即 $\tau = \{\tau_1, ..., \tau_n\}$ 按非增次序排列，其对应的编号为 $J = \{x_1, ..., x_n\}$。

下证该策略的最优性：

反证法：假设 $I \ne J$ 且 $T(I) < T(J)$，其中 $I = \{y_1, ..., y_n\}$，并设 $S = \{v_i | x_{v_i} \ne y_{v_i}\}$ 为 I 与 J 不同位的编号集合，v_i 表示 I 与 J 第 v_i 位不同，且 S 已按 v_i 的非减次序排列。

若 $S = \{v_1, v_2\}$，也即若 I 与 J 只有两位不同，则：

$$\begin{aligned} T(J) - T(I) &= \sum_{i=1}^{n-1} t_{x_i}(n-i) - \sum_{i=1}^{n-1} t_{y_i}(n-i) \\ &= (t_{x_{v_1}} - t_{y_{v_1}})(n - v_1 + (t_{x_{v_2}} - t_{y_{v_2}})(n - v_2) \end{aligned}$$

由于 $x_{v_1} = y_{v_2}, x_{v_2} = y_{v_1}$ *(J 构成到 I 的双射)*，因此上式可写为：

$$(t_{x_{v_1}} - t_{y_{v_1}})v_1 - (t_{x_{v_1}} - t_{y_{v_1}})v_2 = (t_{x_{v_1}} - t_{y_{v_1}})(v_1 - v_2)$$

由于 S 中已按非减顺序排列，且 τ 按非增次序排列，因此 $v_1 < v_2, t_{x_{v_1}} > t_{y_{v_1}}$，该式小于 *0*，这说明了对于解 J，任意调换 *2* 位，$T(J) < T(I)$

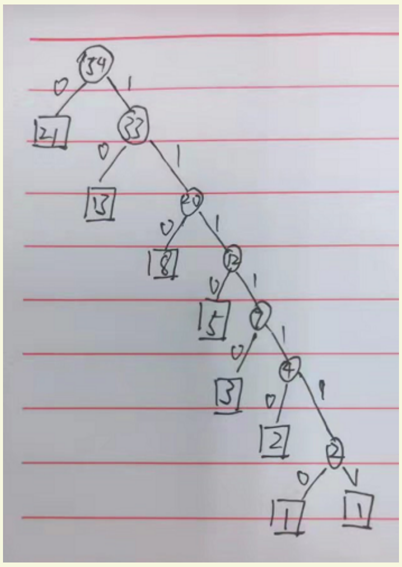
则对于 $S = \{v_1, .., v_m\}$，总可以两两调换使得 J 变为 I，在此过程中，T 不减。

这与假设矛盾，因此 $T(J)$ 为最小值。

Question 2. *2.* 字符 a h 出现的概率分布恰好是前八个 *Fibonacci* 数，它们的 *Huffman* 编码是什么？将结果推广到 n 个字符的频率恰好是前 n 个 *Fibonacci* 数的情况。其中 *Fibon* 数的定义为：$F_0 = 1, F_1 = 1, F_n = F_{n-2} + F_{n-1}$

My Answer 2. 由题：前 *8* 个 *Fibonacci* 数为：*1,1,2,3,5,8,13,21*

则易知，该 *Hauffman* 编码树为：



1 的编码为：*1111111*

2 的编码为：*1111110*

3 的编码为：*111110*

4 的编码为：*11110*

5 的编码为：*1110*

6 的编码为：*110*

7 的编码为：*10*

8 的编码为：*0*

(2) 推广，由 *Fibonacci* 数公式：$F_0 = 1, F_1 = 1, F_n = F_{n-2} + F_{n-1}$，设 $d_n = a_n - \sum_{i=1}^{n-2} a_i$ 则

$$\begin{aligned} d_n &= a_n - \sum_{i=0}^{n-2} a_i \\ &= a_{n-1} + a_{n-2} - \sum_{i=0}^{n-2} a_i \\ &= a_{n-1} - \sum_{i=0}^{n-3} a_i \\ &= d_{n-1} \\ &= d_2 = 2 - 1 = 1 \end{aligned}$$

因此 $a_n - \sum_{i=1}^{n-2} a_i$ 总是大于 *0* 的，且恒为 *1*，则每次完成合并的子树在下一次合并时，总是与当前未合并的序列最小值合并，成为新的子树，则该 *Haffman* 编码树的叶节点总是自底而上，顺序排列 *Fibonacci* 数列，故设算字符 $\vec{a}_n$ 为 n 个 a 相连的字符串，并定义该算字符的加法：$\vec{a}_m + \vec{b}_n$ 前 m 个字符为 a，后 n 个字符为 b 的字符串。

则 n 个 *Fibonacci* 的 *Haffman* 编码为：

第 *1* 个 *Fibonacci* 数的编码为：$\vec{1}_n$

第 *2* 个 *Fibonacci* 数的编码为：$\vec{1}_{n-1} + \vec{0}_1$

第 *3* 个 *Fibonacci* 数的编码为：$\vec{1}_{n-2} + \vec{0}_1$

...

第 $n-1$ 个 *Fibonacci* 数的编码为：110

第 n 个 *Fibonacci* 数的编码为：10

Question 3.

3. 设 $p_1, p_2, \cdots, p_n$ 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 $\sum_{p_i \in Q} a_i \le L$。

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 $\sum_{p_i \in Q} a_i / L$ 取最大值的子集合。

My Answer 3. *(1)* 由题，设 $A = \{a_1, ..a_n\}$，并按非减序列排列。程序是否存放设为 $X = \{x_1, ..., x_n\}, x \in \{0, 1\}$，代长 $P = \{a_1, ..., a_n\}$，并设 $N(V) = \sum_{i=1}^{n} x_i = k$，也即前 k 个最小带长的程序被放入。

设按照上述策略存放的解为 $J = \{x_1, ..., x_n\}$，易知 $\sum_{i=1}^{k} a_i \le L, \sum_{i=1}^{k+1} a_i > L$

假设存在 $I = \{y_1, ..., y_n\}, I \neq J$，使得 $N(I) > N(J), \sum_{i=1}^{m} a_i < L(m > k)$

I 与 J 不存在包含关系。

若 $J \subset I$，则不满足 J 的约束 $\sum_{i=1}^{k} a_i \le L, \sum_{i=1}^{k+1} a_i > L$

$I \subset J$，则不满足 $N(I) > N(J)$ 条件。

则另设 $S_1 = \{i | x_i = 0 \in I \ and \ y_i = 1 \in J\}, S_2 = \{i | x_i = 0 \in J \ and \ y_i = 1 \in I\}$，且由于 $N(I) > N(J), |S_2| > |S_1|$

由于 J 已按不减顺序排列，于是总能将 I 中 S_2 的元素替换成 I 中 S_1 的元素并保持 $\sum_{i=1}^{m} a_i < L$，直到 S_1 全部替换完为止。

此时 I 包含了 J 并且 $N(I) > N(J)$，由上述讨论知与不存在包含关系矛盾，因此 I 不存在

从而 J 即为最优解。

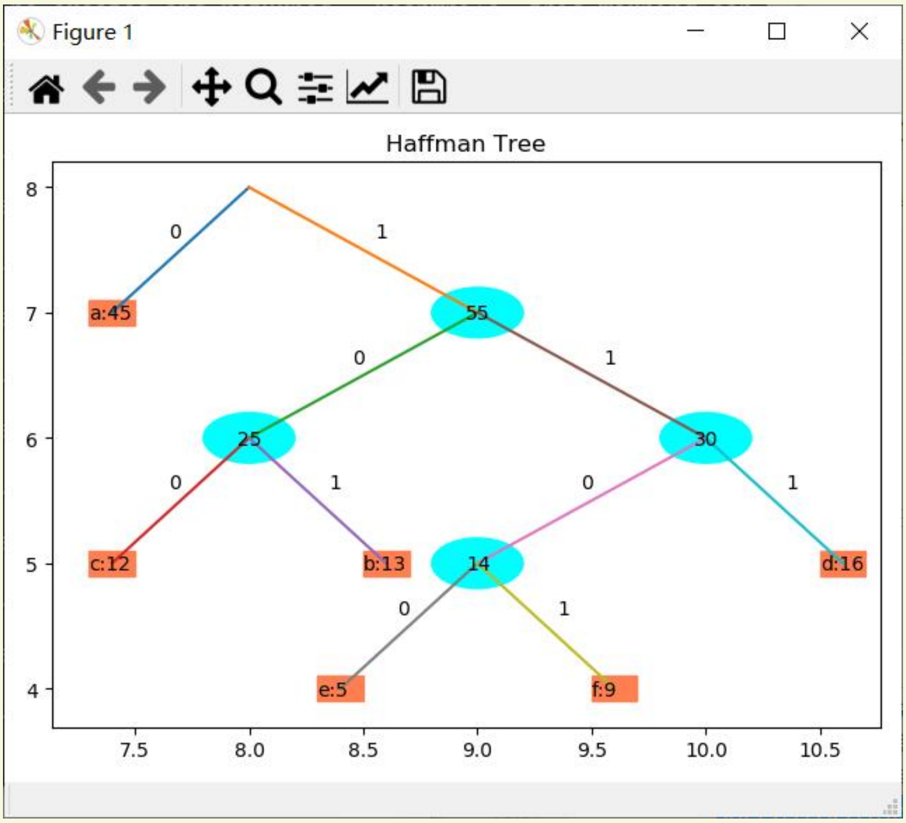
(2) 磁带的利用率为 $\frac{\sum_{p_i \in Q} a_i}{L}$，最小可以减小到 *0*($a_i << L$ 或 $a_i > L$，$\forall i$)。

(3) 例如 $L = 10$，$\{a_1, a_2, a_3, a_4, a_5\} = \{1, 2, 3, 3, 4\}$，若采用上述贪心算法，则 $\sum_{i=1}^{4} = 9$，然而最长的子集合应该是 $\{a_1, a_2, a_3, a_5\}$，长度为 *10*，$\frac{\sum_{p_i \in Q} a_i}{L} = 1$

Question 4. *myHaffman(arr,fre,node)*

My Answer 4. 由题可得以下伪代码:

node myHaffman(arr,fre,node)
  //arr 是 *n* 个字符数组
  //fre 是其对应数频数组
  //node 是当前叶节点与中间节点数组
  if node 只有一个节点 *then*
    return node[0]
  end if
  对 *arr*、*fre*、*node* 按照 *fre* 非减顺序排序，并一一对应
  声明一个新的 *nodeFather* 的子节点指向 *node[0]* 与 *node[1]*
  arr.append(None)
  fre.append(fre[0]+fre[1])
  node 数组添加 *nodeFather*
  del arr[0],arr[1]
  del fre[0],fre[1]
  del node[0],node[1]
  return myHaffman(arr,fre,node)
endmyHaffman
  实现效果如下:



My Answer 5. *(1)*

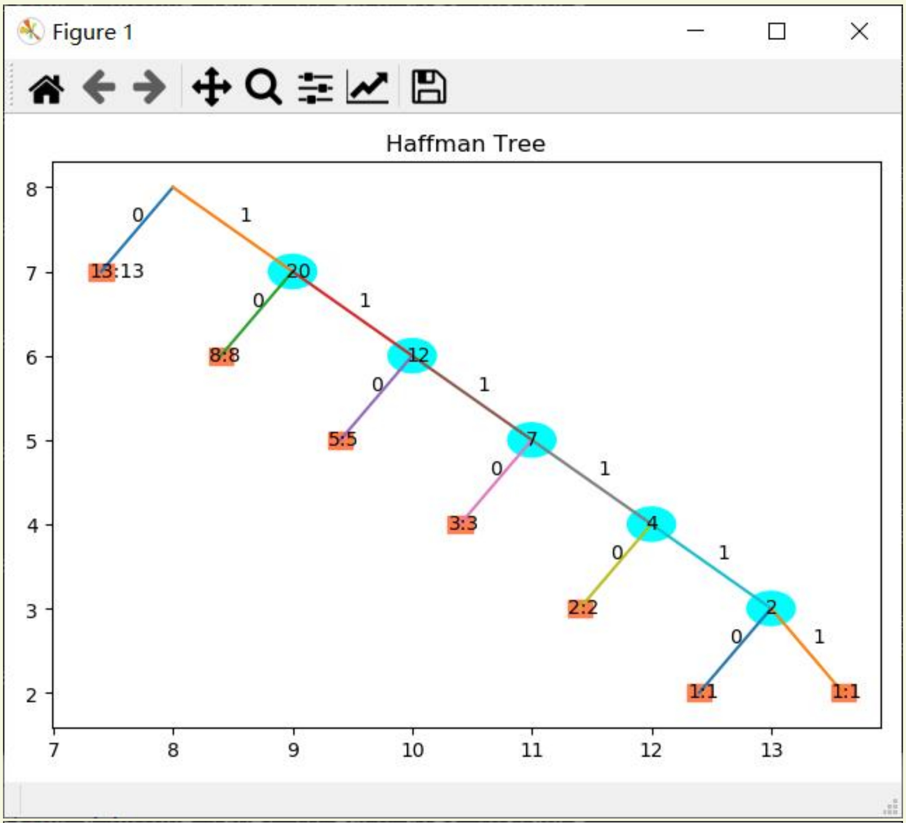


Figure 1
Haffman Tree
0
1
13:13
20
8:8
12
5:5
7
3:3
4
2:2
2
1:1
1:1

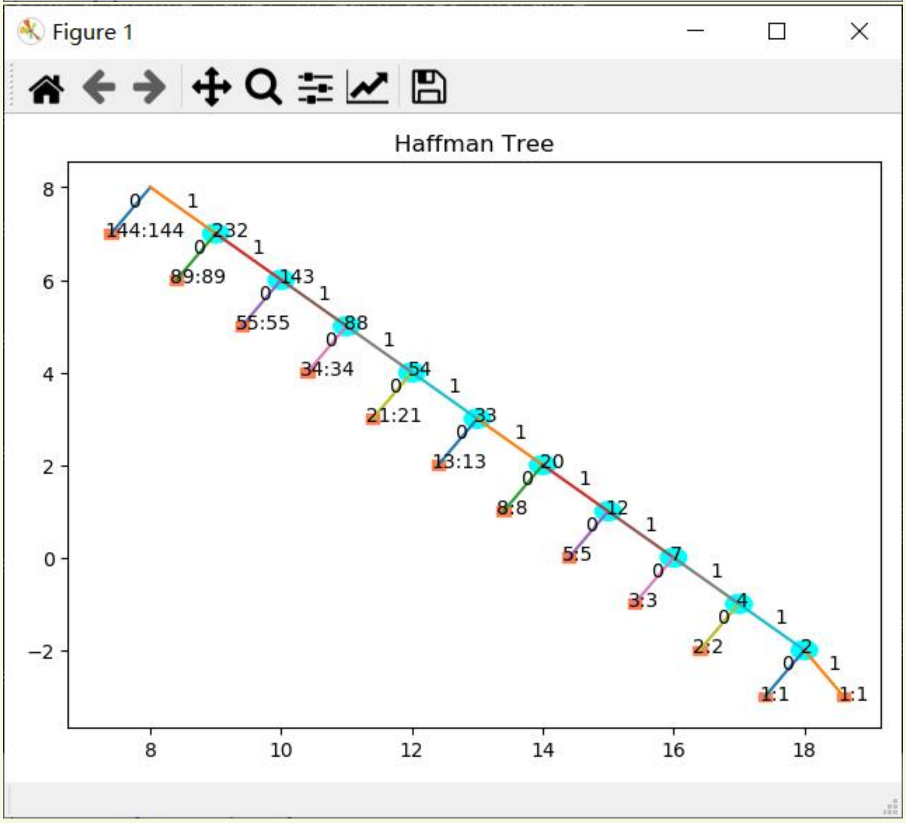


Figure 1
Haffman Tree
144:144
232
89:89
143
55:55
88
34:34
54
21:21
33
13:13
20
8:8
12
5:5
7
3:3
4
2:2
2
1:1
1:1

Question 5. 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 *Krushkl* 算法都能求得最优解。

My Answer 6. *(1)* 设图为 $G = (V, E)$，则该图诱导出的数集合具有拟阵结构 $M = (E, \mathscr{J})$，其中 $\mathscr{J}$ 为所有独立边子集构成的族。易知 E 有限非空，且设赋权函数 $w(x), x \in E$ 为边 x 的权值，$w(A) = \sum_{x \in A} w(x)$。下证交换性质：

设 A 和 B 分别诱导出 G 的两个森林 T_1, T_2，设它们的分支数分别为 k_1, k_2，则 $|A| = n - k_1, |B| = n - k_2$。

由于 $|A| < |B|$，所以 $k_1 > k_2$，这说明 B 中至少有一条边 e 使得其两个端点在 T_1 的不同分支上，这条边不属于 A。于是 $|A| \cup \{e\} \in \mathscr{J}$

(2) 对于 *Prim* 算法，先找到不属于子树的边集合，使得 $A \cup \{x\} \in \mathscr{J}(M)$，再从中找最小的边纳入到原先的子树中；对于 *Kruskal* 算法，先按照赋权函数非减排列不属于子树的边集合，再从中从大到小找到满足 $A \cup \{x\} \in \mathscr{J}(M)$ 的边。

这两种算法增边的准则都要满足两个条件：*a.* 权值最小 *b.* 满足 $A \cup \{x\} \in \mathscr{J}(M)$，因此在执行条件的顺序上不影响算法。

又由 *GreedyMatroid* 算法可知这两种算法都满足 *GreedyMatroid*，因此由定理：设 $M = (S, \mathscr{J})$ 是赋权 w 的拟阵，则算法 *GreedyMatroid(M,w)* 返回的子集 A 是 M 的最优独立集，以及最优生成树问题具有拟阵结构可知，*Prim* 算法和 *Kruskal* 算法均具有最优性