Homework 5

和泳毅 PB19010450

1.假定一个数据文件由 8 位字符组成，其中所有 256 个字符出现的频率大致相同：最高的频率也低 于最低频率的 2 倍。证明：在此情况下，赫夫曼编码并不比 8 位固定长度编码更有效。

答:

记该文件为C。

$\forall a,b\in C$，有$a.freq+b.freq>x.freq,\forall x\in C$。所以在建赫夫曼树的过程中，256个字符被两两组合成128棵树。对它们的根做同样的操作，重复到结束，最后得到一棵深度为$\lg 256=8$的满二叉树。每一个字符都是8位长度的编码，并不比 8 位固定长度编码更有效。

2.令S是一个有限集，$S_1,S_2,\dots,S_k$是S的一个划分，这些集合都是非空且不相交的。定义结构$(S,\mathscr{I})$满足条件$\mathscr{I}=\{A:|A\cap S_i|\leq 1,i=1,\dots,k\}$。证明：$(S,\mathscr{I})$是一个拟阵。也就是说，与划分中所有子集都最多有一个共同元素的集合A组成的集合构成了拟阵的独立集。

答:

设$Y\in\mathscr{I}$，$X\subseteq Y$。则对$\forall i$，有$(X\cap S_i)\subseteq(Y\cap S_i)$。所以$|X\cap S_i|\leq|Y\cap S_i|\leq 1$，即$X\in\mathscr{I}$，$\mathscr{I}$是遗传的。下面证明$(S,\mathscr{I})$满足交换性。

令$A,B\in\mathscr{I}$，且$|A|>|B|$。若：

- **情况1**：对$\forall i$，都有$|A\cap S_i|=|B\cap S_i|=0$。则必然存在$x\in A-B$，$x\cap S_i=\emptyset$，则$|(B\cup\{x\})\cap S_i|=0$，即$B\cup\{x\}\in\mathscr{I}$；
- **情况2**：对$\forall i$，都有$|A\cap S_i|=|B\cap S_i|=1$。则必然存在$x\in A-B$，$x\cap S_i=\emptyset$，则$|(B\cup\{x\})\cap S_i|\leq 1$，即$B\cup\{x\}\in\mathscr{I}$；
- **情况3**：存在j，使得$|A\cap S_j|=1$但$|B\cap S_j|=0$。取$x\in A\cap S_j$，由于S集合间互不相交，$x\cap S_i=\emptyset\ (i\neq j)$。则$|(B\cup\{x\})\cap S_i|\leq 1$，即$B\cup\{x\}\in\mathscr{I}$；
- **情况4**：存在j，使得$|A\cap S_j|=0$但$|B\cap S_j|=1$。不妨设$\forall i\neq j$，没有符合情况3的（否则进入情况3）。接下来分析$A-B$，如果存在$x\in A-B$，$x\cap S_i=\emptyset$，则$|(B\cup\{x\})\cap S_i|\leq 1$，即$B\cup\{x\}\in\mathscr{I}$。如果不存在这样的$x$，即$A-B$中所有元素都至少和一个$S_i$有交集。则根据假设，$\forall x\in A-B$，存在$i$使得$x\cap S_i=1$，即$A\cap S_i=x$，且$|B\cap S_i|=1$（假设不存在情况3），$B\cap S_i\neq x$（不然不属于$A-B$）。所以由于$|A\cap B|+|A-B|=|A|$，$|B|\geq|A\cap B|+|A-B|+1>|A|$，矛盾。所以一定存在满足条件的$x$。

所以$(S, \mathscr{I})$满足交换性，是一个拟阵。

3.$A = a_1, \ldots, a_n$表示一个正整数集合。A中的元素之和为N。设计一个$O(n \cdot N)$的算法来确定是否存在一个 A 的子集 B，使得$\sum_{a_i \in B} a_i = \sum_{a_i \in A-B} a_i$。

答:

采用0-1背包问题思想，n个物品，重量分别为$a_1, \ldots, a_n$，背包承重量为$N/2$。

记数组$dp[i]$表示A中能否取出和为i的子集，数组取值为1或0，1表示可以，0表示不可以。$a[j]$表示第j个元素的值。

递推关系：$dp[i] = dp[i] || dp[i - a[j]]$

伪代码:

```
IF-PAARTITION(a,n,N):
    if(N % 2 != 0) return 0
    sum = N / 2
    let dp[0...sum] be a new table
    dp[0] = 1
    for j = 1 to n
        for i = sum to j
            dp[i] = dp[i] || dp[i-a[j]]
    return dp[sum]
```

时间复杂度为$O(n \cdot N)$。