Homework 5

$S_1 \quad S_2 \cdots S_K$

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。

$A, B \in L$

$B \in L \quad A \subseteq B \Rightarrow A \in L \quad A \oplus B \in L$

2. 令 S 是一个有限集，$S_1, S-2, ..., S_k$ 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 $\mathscr{I} = \{A : |A \cap S_i| \leq 1, i = 1, ..., k\}$。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。

3. $A = a_1, ..., a_n$ 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 $\sum_{a_i \in B} a_i = \sum_{a_i \in A-B} a_i$

1. 设经排序后各字符频率为 $P_1 \quad P_2 \cdots P_{256}$ 其中 $P_1 \leq P_2 \cdots \leq P_{256}$

由赫夫曼编码的构造方式，每次选最小的 $P_i P_j$ 依题意 对 $\forall i, j, k$

$P_i P_j > P_{256} \geq P_k$ 故一直将 P_1 P_{255} 合并直到 P_{255} P_{256}

得到 128 个 $P_i^{(1)} = P_{2i-1} + P_{2i}$

对 $P^{(1)}$ 处理方式与上面相同 且同时满足

对 $\forall i, j, k$ $P_i^{(1)} + P_j^{(1)} > P_{128}^{(1)} \geq P_k^{(1)}$ 故仍是两两合并

最终得到一棵满的二叉树，每个字符都是一个 8 位码

又因为 256 个字符频率大致相同

总二进制位均为 $8 \sum_i P_i$ 与 8 位定长编码相同

2 ① S 是有限集

② 遗传性：设 $B\in\mathcal{I}$ 则 对 $\forall i=1,2,\cdots K$ $|B\cap S_i|\le 1$

$A\subseteq B$ $|A\cap S_i|=|B\cap S_i|-|(B-A)\cap S_i|\le |B\cap S_i|\le 1$

$\Rightarrow A\in\mathcal{I}$

③ 交换性：设 $A\in\mathcal{I}$ $B\in\mathcal{I}$ 且 $|A|<|B|$

(*) 取出所有与 A 有交的 S_i，得 $S_1'\ S_2'\cdots S_l'$ $l<K$

若对 $\forall x\in B-A$ $A\cup\{x\}\notin\mathcal{I}$

取所有 $\{S_i\}$ 的子集 $S'=\{S_i'\}$，s.t. $|A\cap S_i'|=1$ $(i=1,2,\cdots l)$ $l\le K$

则 $|S'|=|A|$

由于 $A\cup\{x\}\notin\mathcal{I}$

故 $\exists\alpha$ s.t. $|\{x\}\cap S_\alpha'|=1$ 即 $x\in S_\alpha'$

$B=(B-A)+(A\cap B)$ (+表直和)

故对 $\forall x\in B$ $\exists\alpha$ s.t. $|\{x\}\cap S_\alpha'|=1$

故 $|B|\le|S'|=|A|$ 等值

$\Rightarrow \exists x\in B-A$ $A\cup\{x\}\in\mathcal{I}$，得证

3. 令 $S[i,j]$ $\begin{cases} 0 & \text{前 } i \text{ 件物品恰好放到一个容量为 } j \\ 1 & \text{前 } i \text{ 件物品恰好放到一个容量为 } j \end{cases}$

对 $i=1,2,\cdots n$ $\quad S[i,0]=1$

对 $j=1,2,\cdots M$ $\quad S[0,j]=0$

$$S[i,j]=\begin{cases} S[i-1,j] & a_i>j \\ S[i-1,j] \text{ or } S[i-1,j-s[i]] & a_i \le j \end{cases}$$

回溯方式 1)若 $S[i,j]=S[i-1,j]$

记 $b[i,j]=$"↑"

2)若 $S[i,j]=S[i-1,j-s[i]]$

记 $b[i,j]=$"↖"

遍历路径时每遇到"↖"就记下 i

直到 $i=0$ 或 $j=0$

只要求 $S[n,\frac{N}{2}]$ 即可

复杂度 $O\left((n+1)(N+1)-1-\frac{N}{2}\right)=O(n\cdot N)$