最佳樹(Optimal Tree)

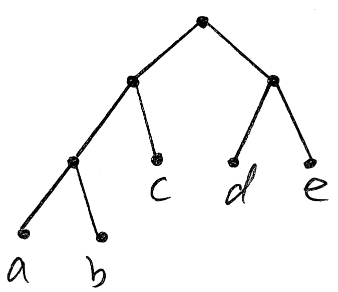
定義:

1. 假設 $T=(V,E)$ 為一個具有 n 個樹葉的 full binary tree,將 n 個權 $w_1, w_2, \ldots, w_n$ 指定給 T 的 n 個樹葉,其中 $0 < w_1 \le w_2 \le \cdots \le w_n$,定義 $W(T)=\sum_{i=1}^{n} w_i \cdot l(w_i)$,稱 T 的權重(weight of T),$l(w_i)$ 指到 w_i 的那個樹葉的階層數(level),也就是路徑長,$i=1,2,\ldots,n$。

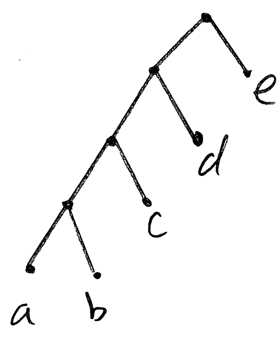
2. 對於指定 $w_1, w_2, \ldots, w_n$ 給 n 個樹葉的 full binary tree 的權有最小的權重,則稱此樹為最佳樹(Optimal Tree)。

例題:計算 weight of Tree

字母	頻率(權重)
a	1
b	2
c	3
d	4
e	5



T_1



T_2

weight of $T_1 \Rightarrow 1\times3 + 2\times3 + 3\times2 + 4\times2 + 5\times2 = 33$

weight of $T_2 \Rightarrow 1\times4 + 2\times4 + 3\times3 + 4\times2 + 5\times1 = 34$

$\therefore$ T_1 有較小的權重。

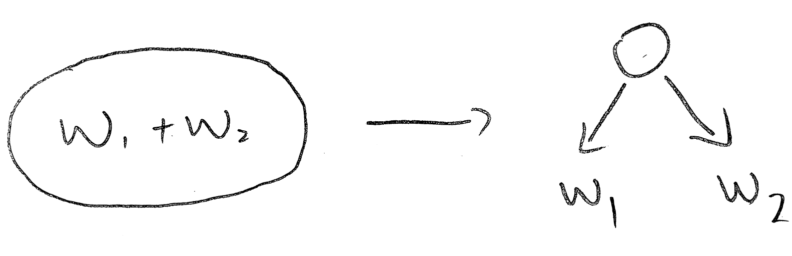
定理：

假設 T 為相對於權 $w_1, w_2, \cdots, w_n$ 的一個最佳樹，其中 $0 < w_1 \le w_2 \le \cdots \le w_n$，則存在另一個最佳樹 T' 滿足有 2 個樹葉指定的權分別為 w_1 和 w_2，使此 2 點互為兄弟，且在最大階層上。

簡單說，最佳樹不唯一，樹的權一樣小啊，最小權的 2 個點會在最下面，且為兄弟。

定理：

假設 T 為相對於 $w_1+w_2, w_3, \cdots, w_n$ 的一個最佳樹，其中 $0 < w_1 \le w_2 \le \cdots \le w_n$，將 T 中具有權 w_1+w_2 的樹葉做修改：



得出另一顆 T_1，則 T_1 為相對於權 $w_1, w_2, \cdots, w_n$ 的最佳樹。

Huffman's algorithm 即這 2 定理創建出最佳樹的方法。

Huffman's algorithm：

由 Huffman's algorithm 建立的 optimal tree 又稱 Huffman's tree。

由例題來解譯 Huffman's algorithm.

例題：

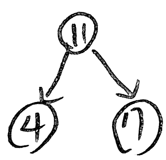
Construct an optimal prefix code following Huffman's recursive procedure for the symbols a, o, q, u, y, z that occur with frequencies 20, 28, 4, 17, 12, 7, respectively.

Ans.

將頻率由小排到大，每次取最小的2個合成一新的點，權重相加，左節點小於右節點。

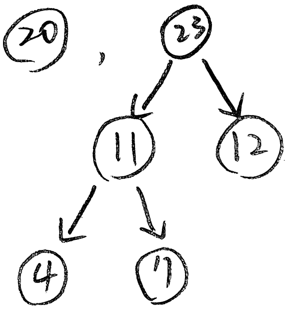
=> 4, 7, 12, 17, 20, 28

1. 取 4, 7 合出新的 11 =>



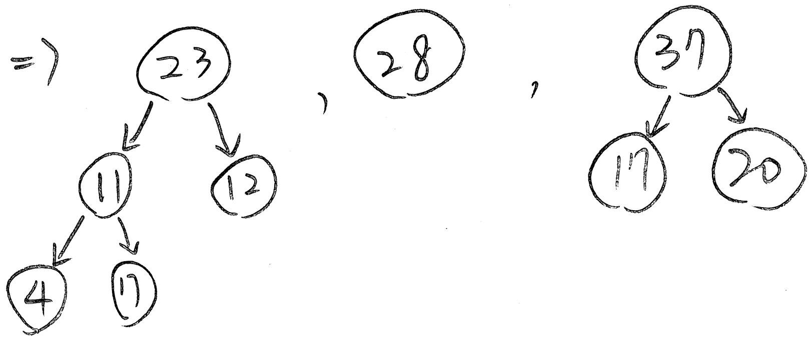
, 12, 17, 20, 28

2. 取 11, 12 合出新的 23 => 17, 20,

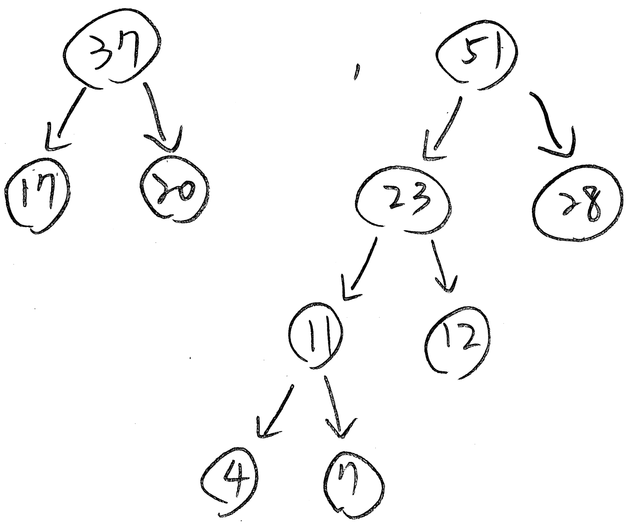


, 28

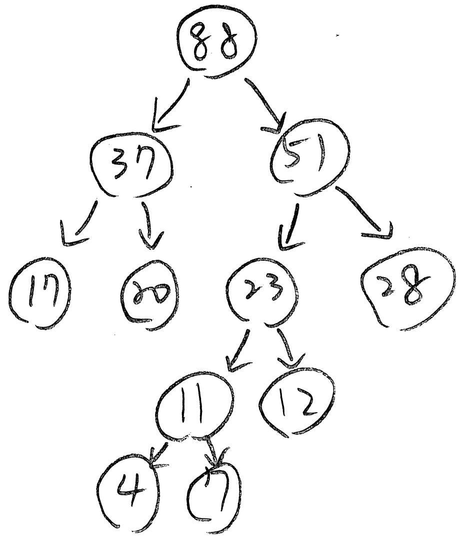
3. 取 17, 20 合出新的 37

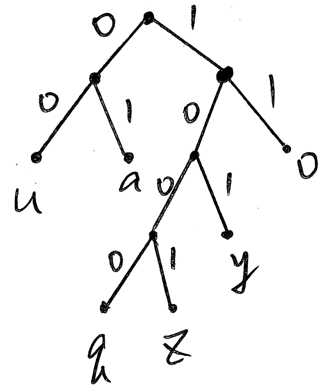
=> 

4. 取 23, 28 合出新的 51.

=> 

5. 取 37, 51 合出新的 88. 即可得出 Huffman's tree.

=> 

=> 

u: 00
a: 01
o: 11
y: 101
z: 1001
q: 1000