Ex 16.3-5

先證明任何一種 prefix code tree 若要 optimal，必定會是 Full binary tree，也就是每個 internal node 一定有兩個 children。

Proof:

假設原來的 prefix code tree T 為 optimal，並且不是 full binary tree，則對任何只有一個 child 的 internal node u（設其 child 為 v），可得下圖：

```
    root
     ...
         \
           p
         /    \
  subtree        u    (only one child v)
                   \
                     v
                    / \
                   /    \
                 -------
            （以v為root的subtree Tv）
```

若將u和v兩個點合在一起，依然能夠維持prefix code tree的特性，因為任何不屬於subtree T_v之leaf所形成的code，必不會是T_v之leaf所形成code的prefix（因為在$edge(p, v)$上的label會造成不同），而T_v之leaf所形成的code本來就是prefix code，所以合併後不會有影響。

又因為合併後T_v之leaf所形成的code變短，所以會比原來的code更好，矛盾。

因為 optimal prefix code tree 為 full binary tree，又有 n 個 leaves，所以會有 n - 1 個 internal nodes 和 $2n$ - 2 條 edges。

表示法：（做法不只一種）

1. 將任何一種 optimal prefix code 以 left edge 標 0，right 標 1 的方式，畫出 prefix code tree 之後，以 DFS 的方式做 tree traversal，每經過一條 edge，就將 edge 上的 label 照順序記下來，總共是 $2n$ - 2bits，如此一來，利用括號配對的方式就可以得知每一組相對的 0 和 1，並藉之將原來的 tree 還原。為了知道何時終止，在後

面加一 bit 的 1 作爲識別。

所以表示 tree structure 的部份共須 $2n - 1$ bits。

2. 然後照每個 leaf 被 traverse 到的順序，將每一個 leaf 代表的 char 依序記錄，由於 char 的範圍是[0 .. n-1]，所以每個 char 須要 ceiling(lg n)個 bits，總共是 n * ceiling(lg n) bits。

3. 總共需要 $2n - 1 + n$ * ceiling(lg n) bits。

Ex. 19.1-3

(a)

用歸納法證明：

在B_k中，任何在depth i的node x都會有$j = k - i$個 1。

Basic：

當$k = 0$ 時的B_0： $j = 0$，label = 0 (0 個'1') => 成立。

Assume：

設 $k = 1 \sim n - 1$ 都滿足在 depth i 的 node x 都會有 $j = k - i$ 個 1。

When：

則$k = n$ 時，對任何在 B_n 中depth i的node x而言：

B_n是由兩個B_{n-1}組合而成

若 x 在left tree中，則在 B_{n-1} 中相對於原本B_n depth i會變成depth $i - 1$，

x 有$(n - 1) - (i - 1) = n - i$ 個 1，又因 left tree 的 label 是加 0，

所以在B_n中，x有$n - i + 0 = n - i$個 1

若 x 在right tree中，則在 B_{n-1} 中depth和原本相同，

x 有$(n - 1) - i = n - i - 1$ 個 1，又因 left tree 的 label 是加 1，

所以在B_n中，x有$n - i - 1 + 1 = n$ - i個 1

由數學歸納法可得証。

(b) 有 j 個 1 的 node，都在 depth $i = k - j$ 的那一層中，

By Lemma 19.1，共有 $C(k, i) = C(k, k - j)$個 k-strings。

(c) 由於B_k中每一個node，都是小binomial tree的root，而在union時，只有root的degree會變化，root下面的其他node，雖然最前面會補一個 0 或 1，但是由於題目說只看最右邊被 0 分開的 1，若不是root至少有一位已經是 0 而隔開了右邊的 1，因此不管前面補 0 還是 1，root以下的點如果原本符合之後還是會符合，故只要考慮B_i的root即可。

同樣用歸納法證明：B_i的root其degree爲i。

在$i = 0$時，B_0的label = 0，沒有'1'，其degree也爲 0 => 成立
在$i = 1$時，由於B_1的root在right tree，所以其Binary code爲 1，
而其 degree 也爲 1 => 成立

假設一直到$i = k - 1$爲止，B_i的root都滿足degree = i。

則在$i = k$時，B_k的root會是right B_{k-1}的root，
而B_{k-1}的root其 label 爲連續$k - 1$個 1，
所以B_k的root其 label 爲 1 + 連續$k - 1$個 1 = k個 1。

由數學歸納法得証。