Yue Li HW 2

No. HW 2

Date 15/09/08

~~Ex~~

28-2.1. .... just follow the algorithm.

Ex

28.2-6. let $P_1 = (a+b)(c-d) = ac - ad + bc - bd$

$P_2 = ad$

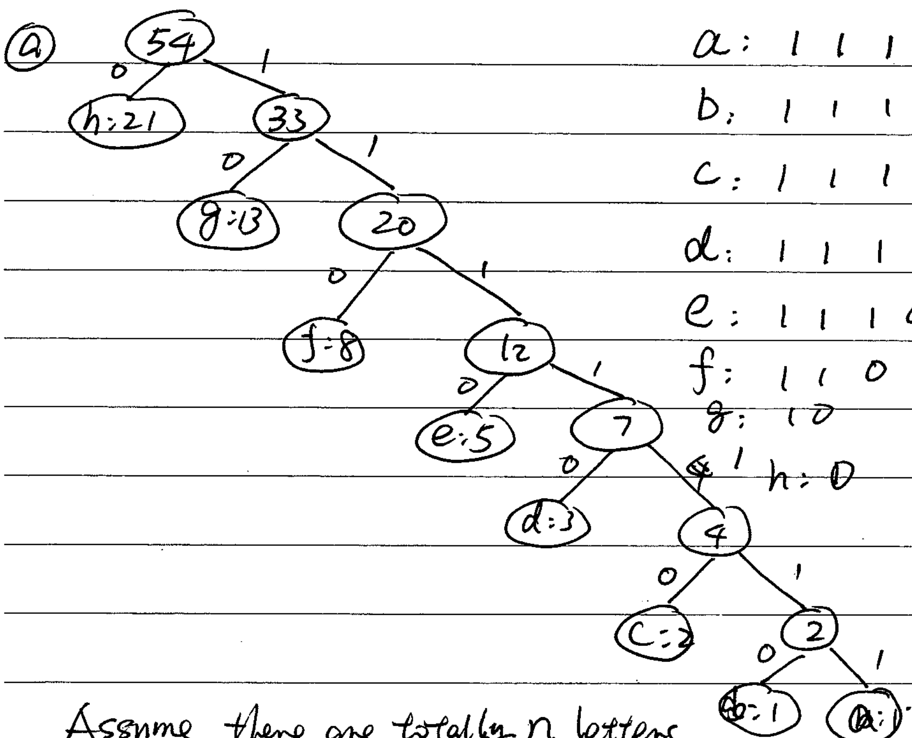
$P_3 = bc$

$\therefore \begin{cases} ac - bd = P_1 + P_2 - P_3 \\ ad + bc = P_2 + P_3 \end{cases}$

only 3 multiplications.

Ex. 16.3-2

(a)



a: 1 1 1 1 1 1 1

b: 1 1 1 1 1 1 0

c: 1 1 1 1 1 0

d: 1 1 1 1 0

e: 1 1 1 0

f: 1 1 0

g: 1 0

h: 0

Assume there are totally n letters,

(b). The letter whose frequence is F_i (the ith Fibonaci No.),

if $i = 1$, then the code will contain $n-1$ "1"s

if $i > 1$, the code will contain, the first $n-i$ "1"s are

and the last one is a "0"s

2.

No.

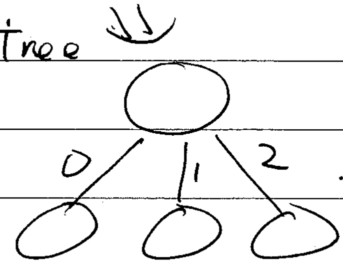
Date

Ex.

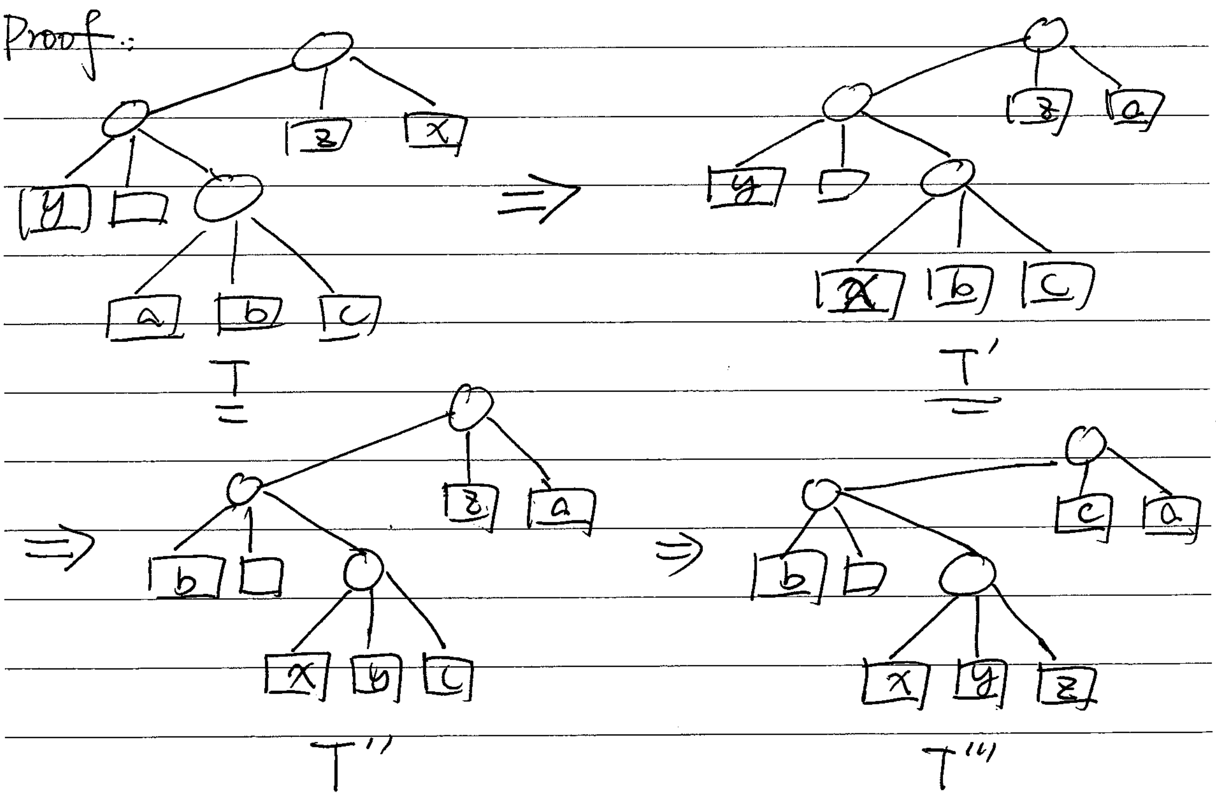
16.3-6

```
HUFFMAN-TENARY(C)
  n ← |C|
  Q ← C
  for i ← 1 to n-1
      do allocate a new node z
         left[z] ← x ← Extract-MIN(Q)
         middle[z] ← y ← Extract-MIN(Q)
         right[z] ← t ← Extract-MIN(Q)
         f(z) ← f(x) + f(y) + f(t)
  return Extract-MIN(Q)
```

TENARY Tree



Proof:

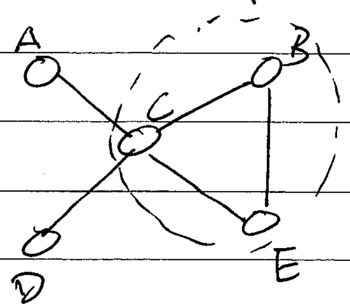


This is to prove $B(T)-B(T') \geq 0$, then we can prove $B(T'')-B(T') \geq 0$

$B(T''')-B(T'') \geq 0$. $B(T)=\sum_{c \in C} f(c) d_T(c)$

Ex. 23.2-8.

A counter-example:



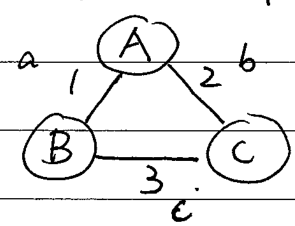
If $V_1 = \{B, C, E\}$, $V_2 = \{A, D\}$.

The Algorithm can never calculate a MST.

Since $E_1 = \emptyset$

Problem 23-4(b)

(a). Obviously, T is not a MST, a counter-example:



The Algorithm may choose b, c edges, thus total weight is $2+3=5$ which is not ~~minimum~~ minimal.

(b)

Implementation: (similar to Kruskal's algorithm)

```
MST-B-Maybe (G, w)
    T ← ∅
    for each vertex v ∈ V[G]
        do MAKE-SET(v)
    for each edge (u, v) ∈ E,
        do if FIND-SET(u) ≠ FIND-SET(v)
            then T ← T ∪ {(u, v)}
                 UNION(u, v)
    return T .
```

No.

Date