

1.

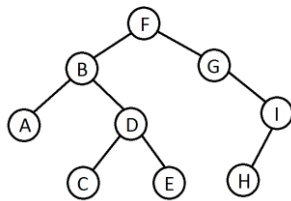
(a) 10, 20, 24, 25, 26, 40, 45, 55

(注意原程式碼與正常的 Quicksort 不太一樣)

(b)  $T(n) = 2T(n/2) + n$ , so,  $T(n) = \theta(n \log n)$

(c)  $T(n) = T(n-1) + n$ , so,  $T(n) = \theta(n^2)$

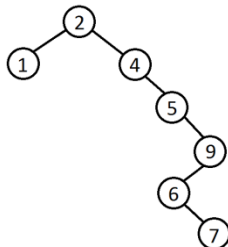
2.



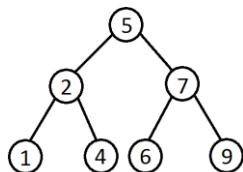
3. (a)  $\theta(n)$  (b)  $\theta(n^2)$  (c)  $\theta(n^2)$  (d)  $\theta(n^5)$  (e)  $\theta(n^4)$

4. (a)

(1)



(2)

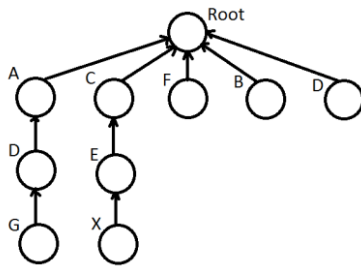


(b)

$$\begin{cases} N(H) = N(H-1) + 1 \\ N_0 = 1, N_2 = 2 \end{cases}$$

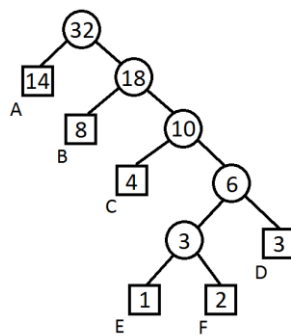
解遞迴式,  $N(H) = F_{H+2} - 1$ , 其中 F 為 Fibonacci number

5.



6. ?

7.



將一串 string 依各 letter 出現的次數或頻率，令其為 set 中的值，每次將 set 中的值最小之兩個取出相加後，形成一個新的 element，並將此值 push 入 set 中，不斷運作直到 set 中值的個數 == 1

此時，Huffman tree 之 leaf 皆為 letter，依其 path 將左子樹編碼為 0，右子樹編碼為 1，即可得 Huffman code，其  $\text{weight} \times \text{path}$  之 code 最小。

8. ?