410530005 馮偉恆 資管三

1、可以用 tail recursion，因為把它recursion 擺在函式的最後一步，讓 call stack 不會被呼叫時創建的新 frame 塞滿，從而避免 stack overflow.

tail recursion：

```
int fib(int n, int a, int b){
    if(n==0)
        return a;
    if(n==1)
        return b;
    return fib(n-1, b, a+b);
}
```

2、(a)

$T(n) = 2T(\frac{n}{4}) + T(\frac{2}{3}n) + cn$

$T(\frac{n}{4})$ $T(\frac{n}{4})$ $T(\frac{2}{3}n)$ $\overset{sum}{\Rightarrow}$ $\overset{cost}{\frac{n}{6}n}$

$T(\frac{n}{16})$ $T(\frac{n}{16})$ $T(\frac{n}{6})$ $T(\frac{n}{16})$ $T(\frac{n}{16})$ $T(\frac{n}{6})$ $T(\frac{n}{6})$ $T(\frac{n}{6})$ $T(\frac{4}{9}n)$ $\overset{sum}{\Rightarrow}$ $\frac{49}{36}n$

$\Rightarrow T(n) \le \sum_{i=0}^{\infty} (\frac{n}{6})^i cn$

$\Rightarrow T(n) \le 1 * \frac{(1-(\frac{n}{6})^{\infty})}{1-\frac{n}{6}} cn$

$\Rightarrow T(n) = O(n)$

(b)

$T(n) = 2T(\sqrt{n}) + \log n$

set $k = \log n$, $n = 2^k$.

$\Rightarrow T(2^k) = 2T(2^{\frac{k}{2}}) + k$

set $S(k) = T(2^k)$

$\Rightarrow S(k) = 2S(\frac{k}{2}) + k$

$\Rightarrow S(k) \le 2S(\frac{k}{2}) + ck$

use Master Theorem $\Rightarrow S(k) = O(k \log k)$

$\Rightarrow T(n) = O(\log n \cdot \log\log n)$ ※

3、∵ heap tree 是二元樹

∴ leaf nodes 最多有 $\frac{n}{2}$ 個

$h = 0, \lceil \frac{n}{2^{0+1}} \rceil = \frac{n}{2}$

$h = k, \lceil \frac{n}{2^{k+1}} \rceil$

$h = k+1, \lceil \frac{\frac{n}{2^{k+1}}}{2} \rceil = \lceil \frac{n}{2^{k+1+1}} \rceil \Rightarrow \lceil \frac{n}{2^{h+1}} \rceil$ 與公式相符 $\Rightarrow$ 由數學歸納法證明此題論述為正確

4、 全部乘以大數，很有可能 overflow

5、 (a) $LCS(i,j,k) = \begin{cases} 0 & \text{if } k=0 \\ LCS(i-1,j-1,k-1)+1 & \text{if } ch_i == ch_j == ch_k \\ \max(LCS(i-1,j,k), LCS(i,j-1,k), LCS(i,j,k-1)) & \text{else} \end{cases}$

Time Complexity : $O(n \times n \times \cdots n) \Rightarrow O(n^k)$ ※

(b) set i = "HELWOR"
j = "HELOW"
k = "HEO"

LCS(i,j,k) = "HEO"

LCS(LCS(i,j),k) = "HE" => 字串 i,j 的最佳解不一定是 i,j,k 的最佳解