Name: Mohamed Kharma

CSC-22000 (Midterm)

Q1) $\lim_{n\to\infty} \frac{g(n)}{f(n)}$

a) $\lim_{x\to\infty} \frac{\log_e x}{\ln x} = 1$

$g(n) \in O(f(n))$

b) $\lim_{x\to\infty} \frac{3x^2 + x^2 \log_2 x}{2000x + \log_2(x!)} = \infty$

$g(n) \in \Omega(f(n))$

c) $\lim_{x\to\infty} \frac{\log_2(x!)}{\ln(1+x^4)^4} = C \leftarrow$ some number

$g(n) \in O(f(n))$

d) $\lim_{x\to\infty} \frac{2^x}{x \log x} = \infty$

$g(n) \in \Omega(f(n))$

e) $\lim_{x\to\infty} \frac{\ln^5(x+x^5)}{\ln^4(x+x^4)} = \infty$

$g(n) \in \Omega(f(n))$

f) $\lim_{x\to\infty} \frac{4^{\log_2 x}}{6x^2} = 1/6$

$g(n) \in \Theta(f(n))$

Q2) $W(n) = n + 4W\left(\frac{n}{8}\right) + 4W\left(\frac{n}{8}\right)$

$\Rightarrow t_n - 4t_{\frac{n}{8}} - 4t_{\frac{n}{8}} = n$

let $n = 8^K$ where $(K = \log_8 n)$, we get:

$\Rightarrow t_K - 4t_{K-1} - 4t_{K-1} = 8^K$ — eq(1)

Replace K with K+1 in eq(1), we get:

$\Rightarrow t_{K+1} - 4t_K - 4t_K = 8^{K+1}$ — eq(2)

Multiply eq(1) by 8, we get:

$\Rightarrow 8t_K - 32t_{K-1} - 32t_{K-1} = 8^{K+1}$ — eq(3)

Subtract eq(2) and eq(3)

$\Rightarrow t_{K+1} - 16t_K + 64t_{K-1} = 0$

Changtis to x^i, we get:

$x^{k+1} - 16x^k + 64x^{k-1} = x^{k-1}(x^2 - 16x + 64) = x^{k-1}(x-8)(x-8) = 0$

→ roots are: $r_1 = 8$

$r_2 = 8$

Therefore, $t_k = C_1 8^k + C_2(k)(8^k)$

$\Rightarrow t_n = C_1 n + C_2(\log n)(n)$

order of $\Theta(n \log n)$

Master theorem:

$a = 8, b = 8, d = 1$

→ $a = b^d \Rightarrow 8 = 8^1$ → case 2

$T(n) = \Theta(n \log n)$

Q3) Inorder:

Method 1: (1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)

Preorder:

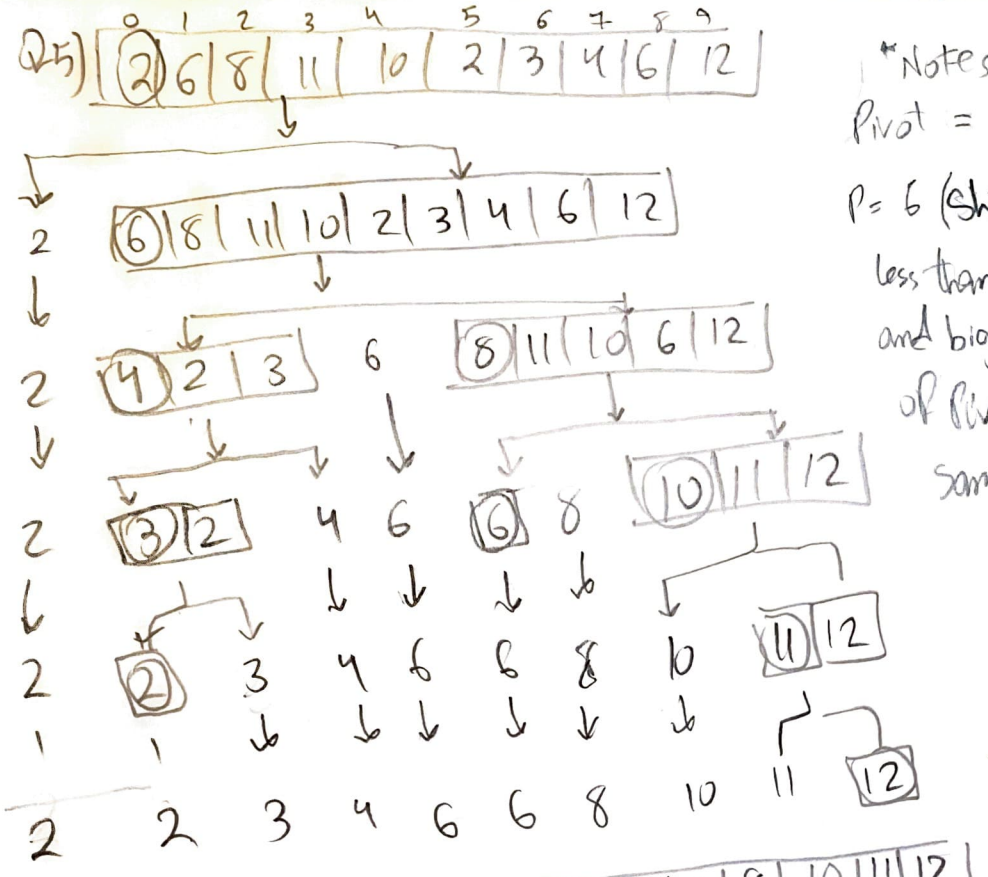
Method 2: (8, 3, 1, 6, 4, 5, 7, 10, 9, 13, 11, 12, 14)

Postorder:

Method 3: (1, 5, 4, 7, 6, 3, 9, 12, 11, 14, 13, 10, 8)

Q4) $\Theta(n^2)$

25)



*Notes:
Pivot = 2 (already sorted)
P = 6 (shifting all elements less than Pivot to left and bigger to the right of Pivot) → do the same for the rest.

Sorted A array: 2 2 3 4 6 6 8 10 11 12

→ Worst case : Θ(n^2)
best case : Θ($n \log n$)
in place : yes
stable : NO, because of swapping

Q5) The code will sort an array using insertion sort. The running time would depend on the input. Best case would be $\Theta(n)$ — array already sorted, worst case $\Theta(n^2)$ - array inversly sorted.

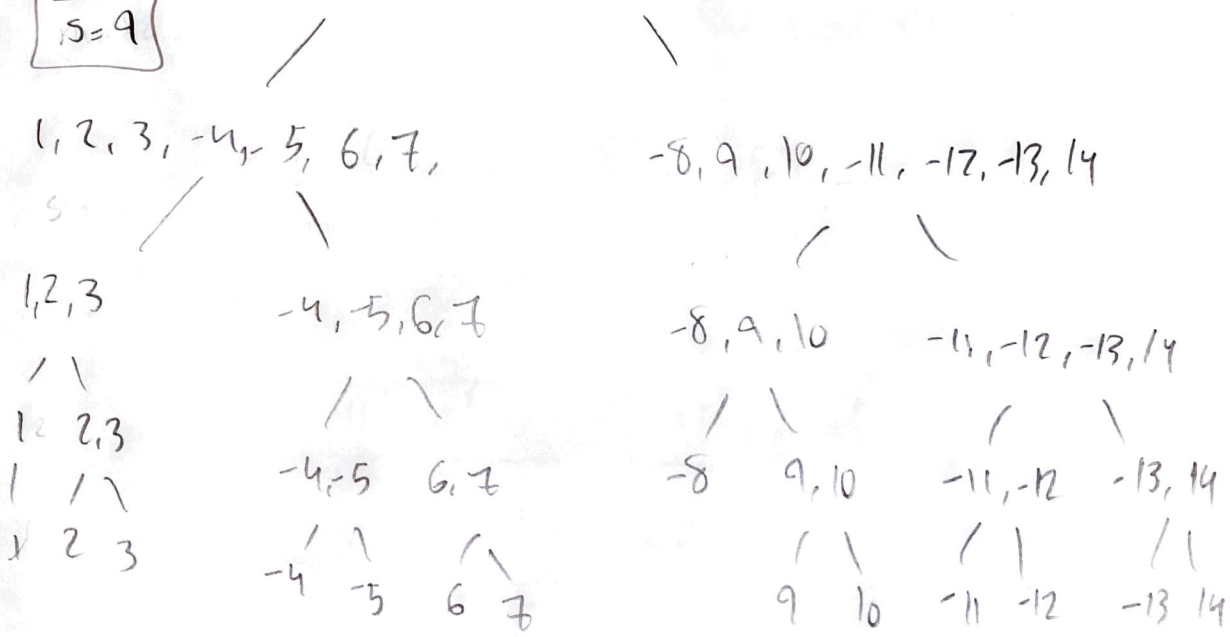
Q6)

- we need to check if a pair of numbers (i, j) are presented in list $L = i + j = a$

1. Sort the list(L) with any sorting Algorithem that takes $\Theta(n \log n)$
2. for every number i in list, we check if $(a-i)$ is also presented. if $a-i$ is found then return true, else we go for the next i number.
3. if no true is returned, then we return False.

Q8) $(1, 2, 3, -4, -5, 6, 7, -8, 9, 10, -11, -12, -13, 14)$

$s = 9$



⇒ Maximum subarray would be 9 and the time complexity is $O(n \log n)$ to get the sum. Using brute force, we get time complexity of $O(n^2)$, therefore divide and conqure would be better to find maximum subarray.