H.W 2

التاريخ: / /

موضوع: _____

① $T(n) = 4T(\frac{n}{2}) + n^3$

by master method

$a = 4$, $b = 2$, $f(n) = n^3$, $n^{\log_2 4} = n^2$

case 1 ~~$f(n) = O(n^{\log_2 4 - \epsilon})$~~

$f(n) < n^{\log_2 4}$ can't be satisfied

case 2 $f(n) = n^{\log_2 4}$ can't be satisfied

case 3 $f(n) > n^{\log_2 4}$ satisfied

$\therefore \Theta(n^3)$

---

② **L1** $T(n) = 8T(\frac{n}{2}) + n^2$ by iterative

$n^2 + 8T(\frac{n}{2})^2 + 8T(\frac{n}{4})$

LevelL2 $= n^2 + 2n^2 + 8^2 T(\frac{n}{4})$

$n^2 + 2n^2 + 8^2(\frac{n^2}{16} + 8T(\frac{n}{8}))$

LevelL3 $= 2^0 n^2 + 2^1 n^2 + 2^2 n^2 + 8^3 T(\frac{n}{8})$

$\frac{n}{2^k}$ $\therefore 2^k = n$ $\boxed{\therefore \log_2 n = k}$

$\sum_{i=0}^{\log n} 2^i n^2$

$= n^2 + \frac{2^{\log n + 1} - 1}{2 - 1}$

$\therefore O(n^3)$

---

(3)

The given code is sorting alg it uses two nested Loops compare elements of Array A and swaps them if need

outer Loop:

for (i=1 to n) do → this [n]

inner Loop:

for (j=n down to i+1) do

The number of iterations in this Loop [n-i]

Total Iterations = $(n-1)+(n-2)+\cdots 1 = \sum_{i=1}^{n-1} i$

$1 = n-1$c

$1c = n-1$

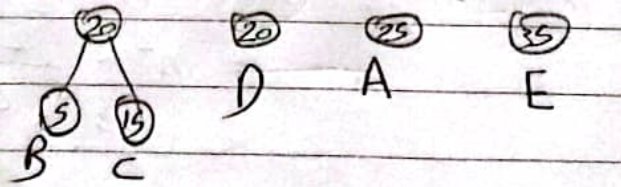
$\therefore$ Time complexity is $O(n^2)$ → $\frac{n(n-1)}{2}$

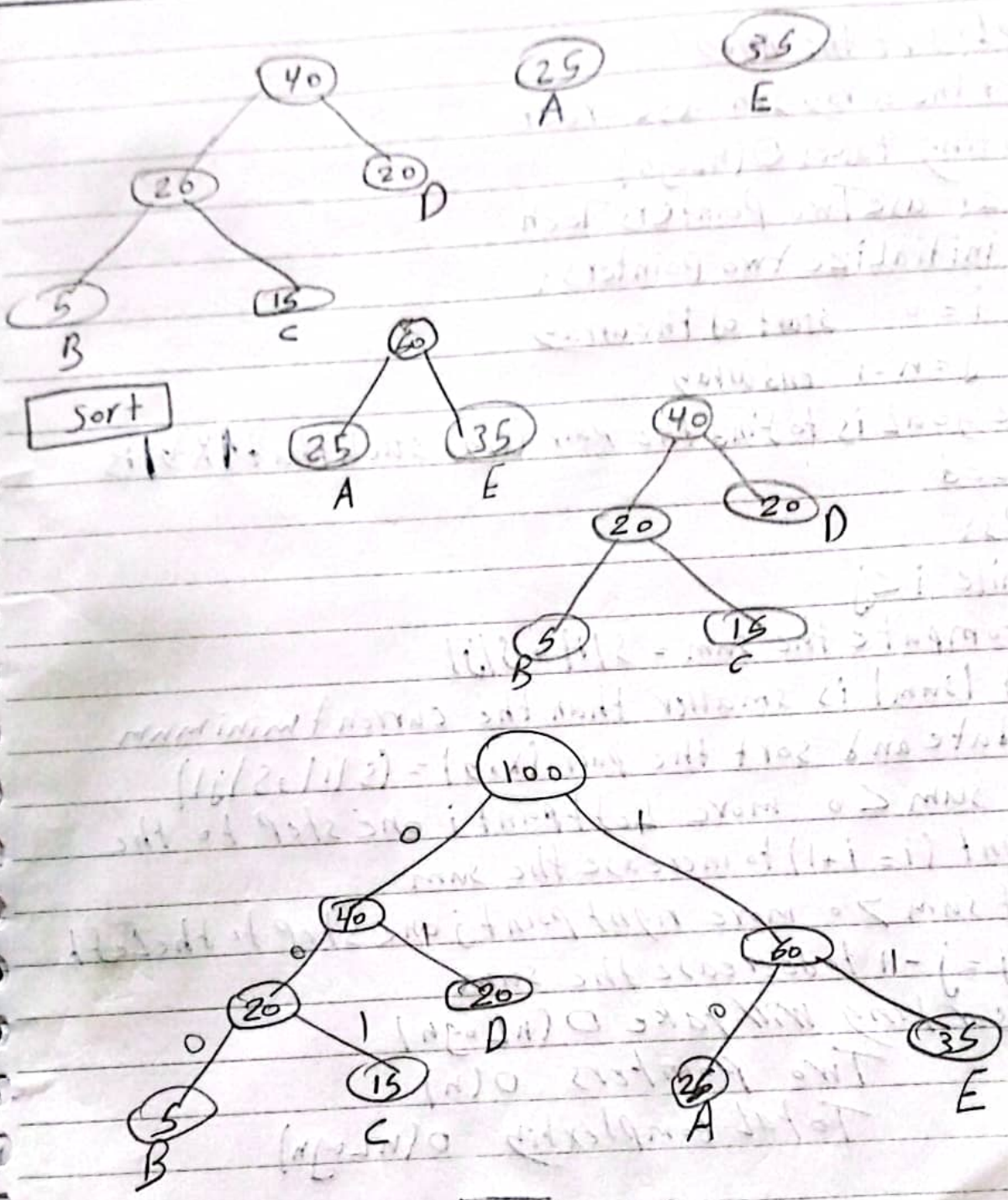
---

(4) Huffman encoding

char	A	B	C	D	E
freq	25	5	15	20	35

Sort Ascending: (5) B, (15) C, (20) D, (25) A, (35) E





	A	B	C	D	E
codeword	01	000	100	10	11
Lenght for each codeword	2	3	3	2	2

التاريخ: / / الموضوع:

(E) step1: sort the Array
sort the array sin asc order
sorting takes O(nLogn)
· step 2: use Two pointers Tech
initialize two pointers:
i = 0 start of the array
j = n-1 end array
The goal is to find the pair (x,y) such that |X-y| is minimized
step3:
While i<j
compute the sum = s[i] + s[j]
if |sum| is smaller than the current minimum
uplate and sort the pair (x,y) = (s[i], s[j])
if sum < 0 move Left point i one step to the right (i = i+1) to increase the sum
if sum > 0 more right point j one step to the Left
j = j-1 to decrease the sum
sorting will take O(nLogn)
Two pointers O(n)
Total complexity O(nLogn)

(F) The weight of the mst of a graph G after two of its edges have their weight changed to $\frac{1}{2}$ can be determined all edges of the graph have a weight of 1 for a graph with n vertices the mst will have n-1 edges each edge have weight 1 the total weight of mst is $\boxed{n-1}$

After two edges are changed to have weight $\frac{1}{2}$

$\therefore$ mst is $2 \times \frac{1}{2} + (n-3) \times 1 = \boxed{n-2}$

$=$ weight of mst