

# Data Structures Final Exam

Name: Hamza Tahir  
Roll No: 18L-1056  
Sec: CS-B

Q1

(a)	$j = 0$	$O(1)$	
	$j < N < 1$	$O(N)$	
	$j++$	$2N$	
	Largest = j	$N$	
	$i = j$	$O(N)$	
	$i < N$	$N(N+1)$	// +1 because to check the last.
	$i++$	$2N(N+1)$	
	If condition	$N(N+1)$	

$$\boxed{\text{Big}(O) = N^2}$$

FindTopPlayers

sort(playerslist, N)  $\rightarrow N^2$

int i	$O(1)$
$i < 3$	3
$++i$	6

$$N^2 + 3 + 9 = N^2 + 9$$

// as we return

$$\boxed{= N^2}$$

Q1

(b)

Construct a max heap for  $n$  number of Players, max will be available at the root.

Second max which is either left or right of root. Remove these nodes and insert a node that has the sum of the largest and second largest into the max heap - Now we will call heapify the second max of this heap is the third max required.

Sort (Player\* Playlist, Int N)

- Construct max heap of Players.
- Root is max.
- Insert it in arr.
- Compare left and right child.
- ~~Greater~~ Greater is second largest.
- Remove largest and second largest  $\rightarrow$  call heapify.
- Largest is 3<sup>rd</sup> largest of max heap, insert in arr. return.

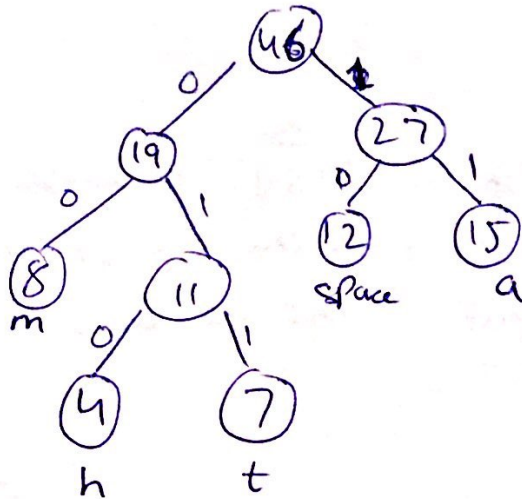


Q3

All codes are wrong.

187-1056

Justify.



$a = 11$

$t = 011$

$m = 00$

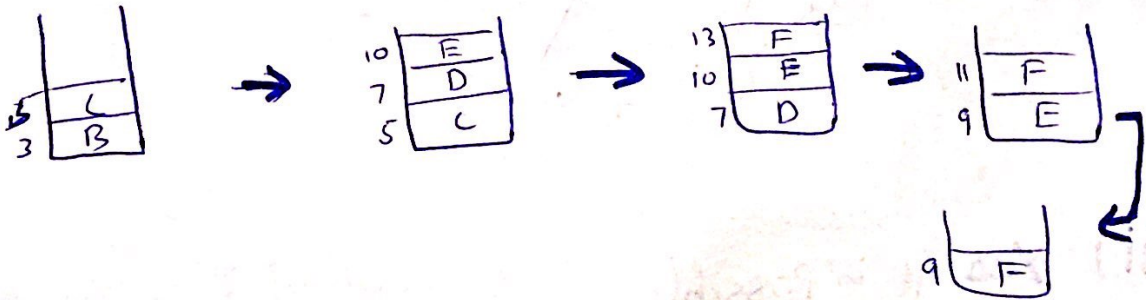
$h = 010$

~~space~~

$space = 10$

Q4

(a) (i)



(ii)

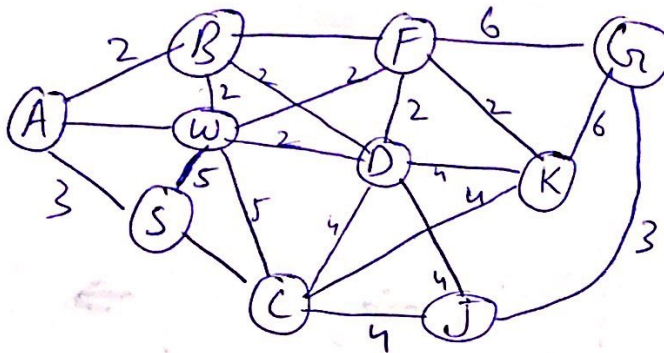
$A \rightarrow 0$   
 $B \rightarrow 3$   
 $C \rightarrow 5$   
 $D \rightarrow 7$   
 $E \rightarrow 9$   
 $F \rightarrow 9$

Q4

(b) No, we can't add edge in max MST because it will invalidate the main criteria of ST. As, edges in MST will become equal to the vertices hence cycle will be formed.

(C)

(i)



(ii) Yes, it is possible.

$AB \rightarrow BF \rightarrow BD \rightarrow BW \rightarrow WD \rightarrow WF \rightarrow DF \rightarrow AS$   
 $\quad \quad \quad 2 \quad \quad \quad 2 \quad \quad \quad 2 \quad \quad \quad 2 \quad \quad \quad 2 \quad \quad \quad 2 \quad \quad \quad 3$

$\rightarrow JG \rightarrow CD \rightarrow DK \rightarrow CK \rightarrow JK \rightarrow JC \rightarrow JD$   
 $\quad \quad \quad 3 \quad \quad \quad 4 \quad \quad \quad 4 \quad \quad \quad 4 \quad \quad \quad 4 \quad \quad \quad 4$

$\rightarrow SW \rightarrow SC \rightarrow WC \rightarrow KG \rightarrow FG \rightarrow KF$   
 $\quad \quad \quad 5 \quad \quad \quad 5 \quad \quad \quad 5 \quad \quad \quad 6 \quad \quad \quad 6 \quad \quad \quad 6$

Q5

181-1056

(a)

CPU Capacity = 1.60 GHz

RAM = 8 gb

Cache

L1 = 256 Kb

L2 = 1 MB

L3 = 6 MB

Virtual memory = 5.2/7.4 GB

(b)

For 10

	<u>Recursive</u>	<u>Iterative</u>
Inorder	1900	45000
Preorder	4200	72400
Postorder	1600	65200

For 50

Inorder	9400	700900
Preorder	13900	557000
Postorder	12200	1098400

For 500

Inorder	25500	1324400
Preorder	74100	1467500
Post order	42000	3502600



Q5

(c)

Big(Oh) notation of recursive and iterative function is  $O(N)$ .

(d)

Iterative function grows faster as compared to recursive function ~~at~~ because of greater space complexity. As space time increases computational time of function also increases.

Q6

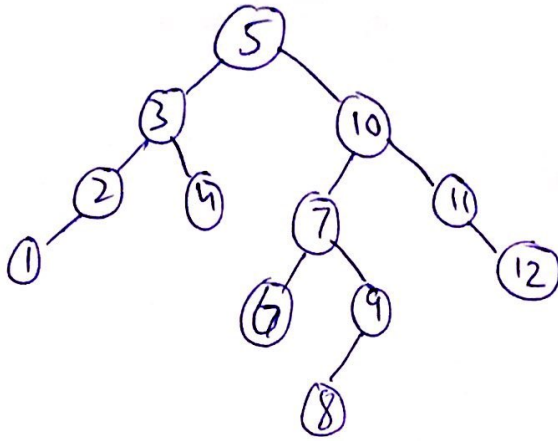
(b)

worst case =  $O(N)$

Average case =  $O(1)$

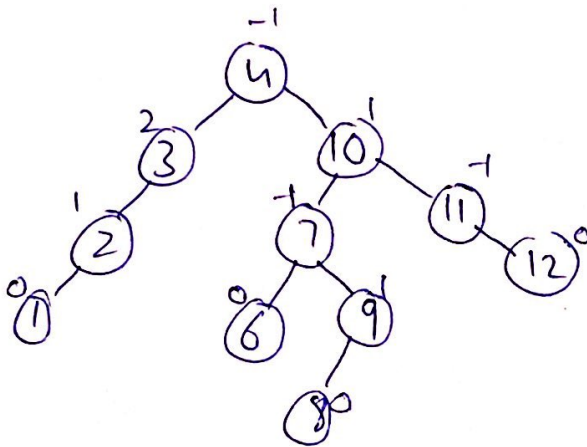
Q7

(a)

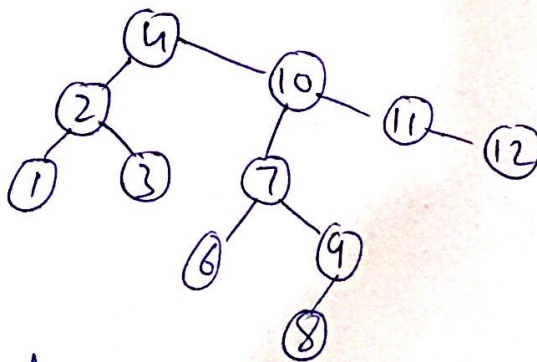


Deleting 5 by replacing with 4 (Predecessor)

New tree is.



(b) First, we will balance 3.



Now we will balance 4.

