7/10

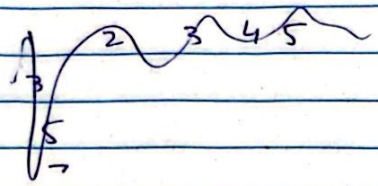
NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE

FAST

Course Name & Section _____ Date _____

Student's Name Dunaiz Waseem Roll No. 24L-7803 Signature Dunaiz

~~n x n Matrix~~



(1)

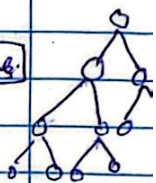
Given: binary heap of size N.

N = 6 , N = 9

Output = 2 , Output = 3

a	b	c	d	e	f	g
0	1	2	3	4	5	6

$\Rightarrow \lg(6) \Rightarrow$ 2 · something



R.W

$2^3 = 8$

$2^2 = 4$

$2^1 = 2$

$2^0 = 1$

a	b	c	d	e	f	g	h	i	j
0	1	2	3	4	5	6	7	8	9

Answer:- $\lfloor \log_2 N \rfloor$ ✓

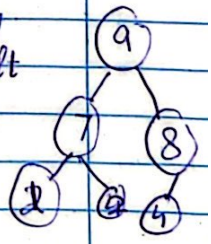
(2)

Heap Sort:

Suppose Given Array:

element:	1	4	2	7	9	8
	0	1	2	3	4	5

Final Result →




```
heapSort (elements)
    // to create heap
    root = elements[0];
    left = (root * 2) + 1
    right = (root * 2) + 2
```

```
Struct Node ( val)
    {int data = val;
    Node* left;
     Node* right;
    };
```

= heapify

heapsort (elements) → sorting algorithm should not be responsible for heap formation

```
{
for(int i=0 ; i<elements.size() ;i++)
    {
    Node n( element[i]);          // to form heap
     n→ left = i*2 +1;
     n→ right = i*2 +2;
    }
```

// after heap is formed, we will apply sorting like reheap down on it.

```
sort(n);
sort(node n){
 parent = element[0];
 left = left = n→ left;
 if (left→ value > n→ value)
    big-index = left→ value;

 if ( right→ value > n→ value)
    big-index = right→ value);

 if ( n→ value != big-index)   (value)
 {swap( n, n→value , big-index);   (value)
    sort ( big-index) ;
 }
```

recursive call to heapify function will reinitialize the heap

wrapper of heapify to actually sort the given array?

heap formation

heap Sorting