

Name : Harishchandrasing Sababsing Kushwah
 Roll No : 296245
 sub : DDC Assignment

Q1

\Rightarrow

Logarithmic growth

n	$n^2 \log(n)$	$n \log(n)$	$10 \cdot n^2$	$\log(n)$	2^n	$n^3 \log(n)$
2	4	2	40	1	4	8
4	32	8	160	2	16	128
8	192	24	640	3	256	1536
16	1024	64	2560	4	65536	16384
32	5120	160	10240	5	4.2949×10^9	163840
2048	46137344	225284199304	111	undefind	9.44×10^{10}	

From above table we can observe that growth of the functions in ascending order

$$\log(n) < n \log(n) < 10n^2 < n^2 \log(n) < n^3 \log(n) < 2^n$$

Q. 2

ROLL NO = 246245

$$x = \text{ROLL-NO} \% 4$$

$$x = 246245 \% 4 = 1$$

$$A = 1, 2, 4, 1, 2, 1, 1$$

$13 = 4$ --- maximum element

A	1	2	4	1	2	1	1
	1	2	3	4	5	6	7

--- $n = 7$

i	4	2	0	1			
	1	2	3	4			

--- counting array

i	$\text{c}[A[i]]$	$c[A[i]]$
1	1	$0 + 1 = 1$
2	2	$0 + 1 = 1$
3	4	$0 + 1 = 1$
4	1	$1 + 1 = 2$
5	2	$1 + 1 = 2$
6	1	$2 + 1 = 3$
7	1	$3 + 1 = 4$

pre sum of counting array

c	4	6	6	7
---	---	---	---	---

B:	1	1	1	1	2	2	4
	1	2	3	4	5	6	7

external array
to store sorted
elements

i	A[i]	crash	B[C[A[i]]]
7	1	4	B[4] = 1
6	1	3	B[3] = 1
5	2	6	B[6] = 2
4	1	2	B[2] = 1
3	4	7	B[7] = 4
2	2	5	B[5] = 2
1	1	1	B[1] = 1

Q.3

$$\Rightarrow 2n^2 + 3n + 1 = O(n^2)$$

$$n \quad f(n) = 2n^2 + 3n + 1 \quad O(n^2) = 3 \cdot n^2$$

0	1	2	3	4	5
1	4	9	16	25	36
2	15	28	45	64	81
3	28	45	64	81	100
4	45	64	81	100	121
5	64	81	100	121	144

$$2n^2 + 3n + 1 = O(n^2) \quad \forall n > 4 \quad \text{and } c = 3$$

Q 4

Initial =

A =	1	2	1	0
	0	1	2	3

pivot = 1 1010 = 0 high = 3 13 = high = 3

i = 3

13 = 3 if (arr[3] > pivot) { → false
13 > 1 0 > 1

i = 2

13 = 3 if (arr[2] > pivot) { → false
13 > 1 1 > 1

i = 1

13 = 3

if (arr[1] > pivot) { → true

13 > 1 2 > 1 → true

swap (arr[1], arr[13])

13 =

A =	1	0	1	2
	0	1	2	3

i = 0

13 = 2

if (arr[0] > pivot) { → false
1 > 1

swap(arr[i], arr[i+1])

$$A = \begin{bmatrix} 1 & 0 & 1 & 2 \end{bmatrix}$$

0 1 2 3

New pivot index = $i = 1$ for next iteration

$i = 2$... from previous iteration

$$A' = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

0 1

$$A'' = \begin{bmatrix} 2 & 1 \end{bmatrix}$$

3

low = 0 high = 1

here low < high

so further algorithm work: adj. min. & add. of

pivot = 1

$i = \text{high} = 1$

$i = 1$

$i = 1$ if($arr[i] > \text{pivot}$) \leftarrow FALSE

$0 > 1$

}

$i = 0$

$i = 1$ if($arr[i] > \text{pivot}$) \leftarrow FALSE

}

swap($arr[low]$, $arr[i]$)

$$A' = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

0 1

New pivot index = $k = 1$

$$A''' = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$low = 0 \Rightarrow high = 0$

$low < high \rightarrow \text{false}$

Now final array after sorting

$$A = \begin{bmatrix} 0 & 1 & 1 & 2 \end{bmatrix}$$

In above array the position of similar elements remains same therefore quick sort is stable sort algorithm.

Q.5

$\Rightarrow n = 3$ items \rightarrow 3 (longer time)

$m = 13$

$IN = (9, 6, 5)$

$P = (15, 18, 20)$

To find an optimal solution find the profit is to weight (P/w) ratio and then apply the greedy approach

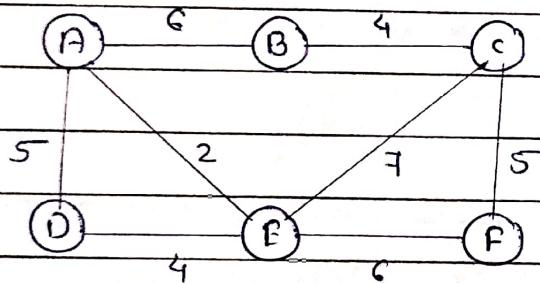
$$P/w = (15/9, 18/6, 20/5)$$

$$= (1.66, 3, 4)$$

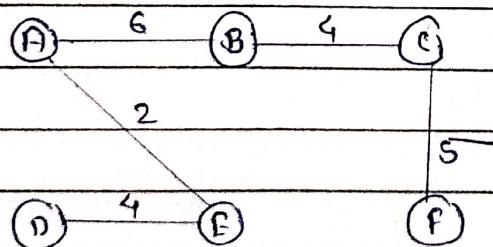
select maximum profit

item	Profit	Inweight
j_1	10	$10 \times 0.1 = 1.0$
j_3	20	$13 - 5 = 8$
j_2	6	$(8 - 2) = 6$
j_4	$(3g) \times 15 = 3.33$	$12 - 2 = 10$
Total profit	29.33	

Q. 6



Q. 1 Kruskals algorithm



(A, E)	(D, E)	(B, C)	(A, D)	(C, F)	(A, B)	(E, F)	(E, C)
2	4	4	5	5	6	6	7

u, v	j, k	i	A	B	C	D	E	F	min cost
		0	A	B	C	D	E	F	0
(A, E)	(A, F)	1	A	B	C	D	A	F	0+2 = 2
(D, E)	(D, F)	2	A	B	C	A	A	F	2+4 = 6
(B, C)	(B, F)	3	A	B	B	A	A	F	6+4 = 10
(A, D)	(A, A)	3	A	B	B	A	A	F	10+0 = 10
(C, F)	(B, F)	4	A	B	B	A	A	B	10+5 = 15
(A, B)	(A, B)	5	A	A	D	A	A	A	10+0 = 10
(E, F)	(A, A)	6	A	A	D	A	A	A	21
(F, C)	(D, A)	7	A	A	A	D	A	A	21
									21

b) prims algorithm

(1) - 5 - (6) - 2 - (1)

(2) - F - 3 - (3)

(3) - - (7) - (6)

(3) - 2 - (6) - 3 - (7)

(1) - 3 - (6) - (4)

Q. 7

$$\Rightarrow \text{ROLL-NO} = 296245$$

$$x = (\text{ROLL-NO} + 50) \div 10$$

$$x = (296245 + 50) \div 10$$

$$x = 5$$

Frequencies of message $\{3, 5, 9, 11, 15, 19, 25\}$

Sort the frequencies in ascending order

$3, 5, 9, 11, 15, 19, 25$

$8, 9, 11, 15, 19, 25$

$11, 15, 17, 19, 25$

$17, 19, 25, 26$

$25, 26, 36$

$36, 51$

Huffman code

Frequencies

$$3 \rightarrow 1000$$

$$5 \rightarrow 1001$$

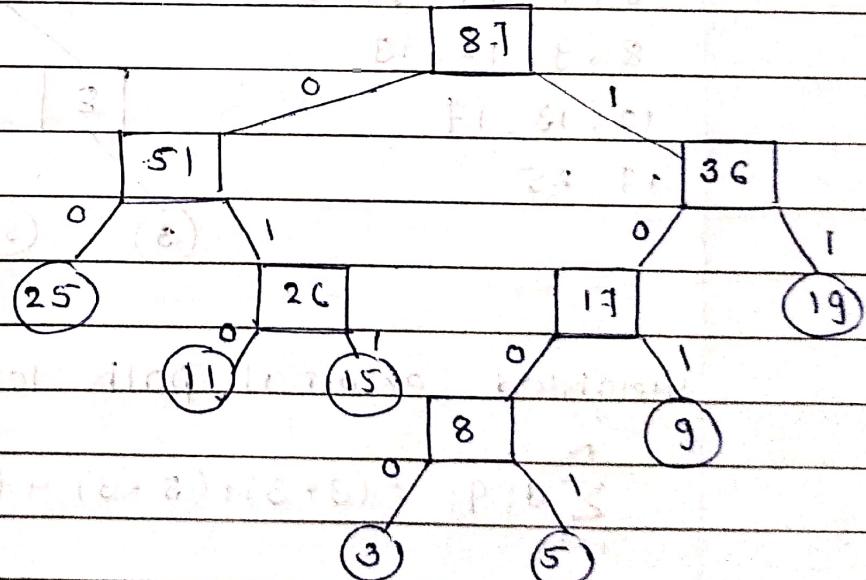
$$9 \rightarrow 101$$

$$19 \rightarrow 11$$

$$15 \rightarrow 011$$

$$11 \rightarrow 010$$

$$25 \rightarrow 00$$



Q. 8

$$\Rightarrow \text{ROLL-NO} = 246245$$

$$x = (\text{ROLL-NO} + 50) \mod 10$$

$$x = (246245 + 50) \mod 10$$

$$x = 5$$

$$\text{Files} = (3, 15, 9, 19, 11, 25, 8, 13)$$

$$\text{Files} = (3, 7, 5, 9, 5, 13)$$

sort in ascending order

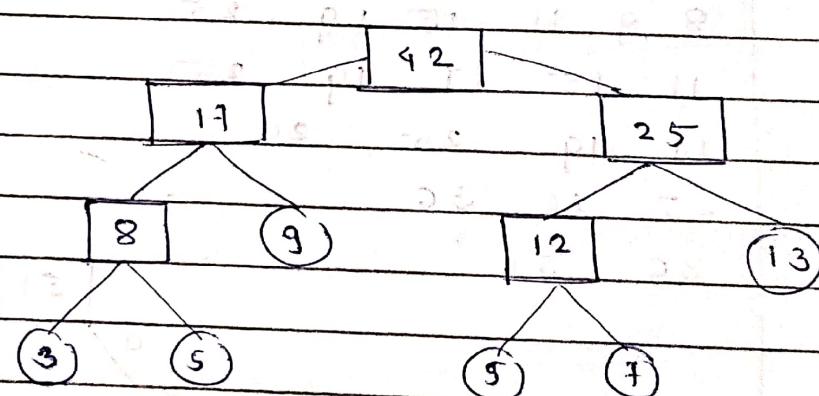
$$3, 5, 5, 7, 9, 13$$

$$5, 7, 8, 9, 13$$

$$8, 9, 12, 13$$

$$12, 13, 17$$

$$17, 25$$



Weighted external path length

$$\sum_{i=1}^n d_i q_i = (3*3) + (5*3) + (9*2) + (5*3) + (7*3) + (13*2)$$

$$= 9 + 15 + 18 + 15 + 21 + 26$$

$$= 104$$

Q. 9

\Rightarrow for $i := 1$ to n do (1)

for $j := \frac{n}{2}$ to 1 do $n + (\frac{n}{2})$

 write(i) $n + (\frac{n}{2})$

$$= n + n + (\frac{n}{2}) + n + (\frac{n}{2})$$

$$= n + \frac{2 + n + n}{2}$$

$$= n + n^2$$

$$= n^2 + n$$

in asymptotic notation $f(n) = n^2 + n = O(n^2)$

Q. 10

\Rightarrow Algorithm Exponent (a, n)

{

$p = 1$; (1)

for $i := 2$ to n do $(n-2+1+1) = n$

$p = a * p$; $(n-1)$

return p ; (1)

{

$$= 1 + n + n - 1 + 1$$

$$= 2n + 1$$

8-11

Algorithm No	Time complexity (Best)	Time complexity (Average)	Time complexity (Worst)	Space complexity	Stable	In-place	Best case input	Worst case input
1. Bubble sort	$O(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$	Yes	Yes	Already sorted	Reverse sorted
2. Selection sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$	No	Yes	-	Any unsorted input
3. Insertion sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$	Yes	Yes	Already sorted	Reverse sorted
4. Merge sort	$\Omega(n \log n)$	$\Theta(n \log n)$	$O(n \log n)$	$O(n)$	Yes	No	-	Any unsorted input
5. Quick sort	$\Omega(n \log n)$	$\Theta(n \log n)$	$O(n^2)$	$O(\log n)$	No	Yes	Randomly sorted	Already sorted
6. Heap sort	$\Omega(n \log n)$	$\Theta(n \log n)$	$O(n \log n)$	$O(1)$	No	Yes	-	Any unsorted input
7. COUNTING sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$	Yes	No	small range of k	large range of k
8. RADIX sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$	Yes	No	small digit size	large digit size
9. BUCKET sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$	Yes	No	uniform data	skewed data