solution of set - 1

① Write down an algorithm to delete an element from the queue (Deque).

Step1: Check if the queue is empty

Step 2: If the queue is empty produce underflow and exit

Step 3: If the queue is not empty access the data where front is pointing

Step4: Increment front pointer to point to the next available data element.

Step 5: Return success

OR

Step1: If front =-1 and rear =-1 display queue is empty (underflow) and exit

Step2: Else data = queue [front]
front = front+1

Step 3: Stop

Q no.2 ans Prove by mathematical induction that:

$1+3+5+\dots+(2n-1)=n^2$

Soln

Let $P(n) = 1+3+5+\dots+(2n-1)=n^2$

1. Basic step: for $n=1$, we have
$P(1) = 1 = (1)^2$, hence $P(1)$ is true.

2. Induction Hypothesis:
Assume $P(k)$ is true i.e.
$P(k) = 1+3+5+\dots+(2k-1) = k^2$

3. Inductive step: Now, we wish to show $P(k+1)$ is true. So, adding $P(n)$, then
$1+3+5+\dots+(2k-1)+(2k+1)=(n+1)^2$
$\therefore P(n+1)=(n+1)^2$, is true.

Thus, by mathematical induction $P(n)$ is true for all n

3) Howmany numbers must be selected from the {1, 3, 5, 7, 9, 11, 13, 15} to guarantee that at least one pair of these numbers add upto 16? solve using pigeonhole principle.

→ The pairs of numbers that sum 16 are: (1, 15), (3, 13), (5, 11), (7, 9) i.e. 4 pairs of numbers are there that add to 16. If we select 5 numbers, then by pigeonhole principle there are at least.

ceil (5/4) = 2 numbers, that are from the set of selected 5 numbers, that constitute a pair. Hence, 5 numbers must be selected.

4) Write down algorithm for binary search.

1. Start
2. Read the search element from the user
3. Find the ~~element~~ middle element in the sorted list.
4. Compare the search element with the middle element in the sorted list.
5. If both are matching then display "given element found!!!" and terminate the function.
6. If both are not matching, then check whether the search element is smaller or larger than middle element
7. If the search element is smaller than middle element, then repeat steps 2, 3, 4 and 5 for the left sublist of the middle element.
8. If the search element is larger than middle element, then repeat step 2, 3, 4 and 5 for the right sublist of the middle element
9. Repeat the same process until we find the search element in the list or until sublist contains only one element

10. If that element also doesn't match with the search element, then display "Element not found in the list!!!" and terminate the function.

11. Stop.

(5) sort the following elements by using insertion sort. A = {4, 3, 1, 10, 2, 16, 8}

Array position	0	1	2	3	4	5	6
Initial State	4	3	1	10	2	16	8
Pass 1	4	3	1	10	2	16	8
Pass 2	3	4	1	10	2	16	8
Pass 3	1	3	4	10	2	16	8
Pass 4	1	3	4	10	2	16	8
Pass 5	1	2	3	4	10	16	8
Pass 6	1	2	3	4	10	16	8
Pass 7	1	2	3	4	8	10	16.

Sorted array is = {1, 2, 3, 4, 8, 10, 16}

(6) In a lottery, there are 10 prizes and 25 blanks. A lottery is drawn at random. What is the probability of getting a prize?

ans: 2/7

Explanation:

Total number of outcomes possible, $n(S) = 10 + 25 = 35$.

Total number of prizes, $n(E) = 10$.

$$P(E) = \frac{n(E)}{n(S)} = \frac{10}{35} = \frac{2}{7}$$

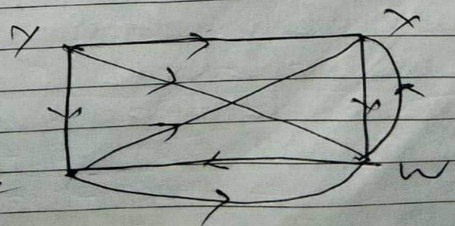
(7) Let $A = \{1, 2\}$ and $B = \{3, 4, 5\}$. Construct $A \times B$ and $B \times A$ and then comment.

Soln

$A \times B = \{(1,3)\ (1,4),\ (1,5)\ (2,3)\ (2,4),\ (2,5)\}$

$B \times A = \{(3,1), (3,2)\ (4,1)\ (4,3)\ (5,1)\ (5,2)\}$

(8) Find the adjacency matrix to represent the directed graph shown in figure where vertices are ordered as $V_1 = X$, $V_2 = Y$, $V_3 = Z$ and $V_4 = W$.



8 Ans: The adjacency matrix of the directed graph is:

$$A = \begin{array}{c c} & \begin{array}{cccc} x & y & z & w \end{array} \\ \begin{array}{c} x \\ y \\ z \\ w \end{array} & \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

set solution set -2

① Write down an algorithm for push operation in stack.

Let STACK[MAXSIZE] is an array to implement the stack. The variable top denotes the top of the stack. This algorithm adds or inserts an item at the top of the stack.

1. Start
2. Check for stack overflow as
   if top = MAXSIZE-1 then
   Print "stack overflow" and Exit the Program
   else Increase top by 1 as,
   Set, top = top + 1
3. Read elements to be inserted say element
4. Set stack [top] = element // Insert item in new position.
5. Stop.

② Solve the recurrence relation $a_n = a_{n-1} + 2$ subject to initial condition $a_1 = 3$.

~~Sol~~ⁿ

$a_n = a_{n-1} + 2$ step 0

$= (a_{n-2} + 2) + 2$ step 1

$= (a_{n-3} + 2) + 4$ step 2

$= (a_{n-4} + 2) + 6$ step 3

$= (a_{n-5} + 2) + 8$ step 4

⋮

$a_n = a_n + 2(n-1)$ step (n-1)

(3) Write down expansion of $(1+y)^6$ using Pascal's triangle theorem.

Sol^n^/

Here, $n=6$, so use Pascal's triangle upto $n=6$

When $n=0$							1							
When $n=1$						1		1						
When $n=2$					1		2		1					
When $n=3$				1		3		3		1				
When $n=4$			1		4		6		4		1			
When $n=5$		1		5		10		10		5		1		
When $n=6$	1		6		15		20		15		6		1	

$$\therefore (1+y)^6 = 1(1)^6 + 6(1)^5 y + 15(1)^4 (y)^2 + 20(1^3)(y)^3 + 15(1)^2 (y)^4 + 6(1)^1 (y)^5 + 1(1)^0 (y)^6$$

$$= 1 + 6y + 15y^2 + 20y^3 + 15y^4 + 6y^5 + y^6$$

(4) ~~Write down~~ an algorithm ~~for Breadth first search.~~

(4) Write down the steps for Kruskal's algorithm.

Step 1: Start

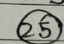
Step 2: List all the edges of G with non-decreasing order of their weights.

Step 3: Select an edge of minimum weight (if there are more then one edge of minimum weight, arbitrarily choose one of them). This will be the first edge of T.

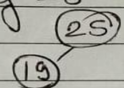
Step 4: At each stage, select an edge of minimum weight from all the remaining edges of G if it doesn't form a cycle with the previously selected edges in T. then add the edge to T.
Step 5: Repeat step 3 until n-1 edges have been selected.
Step 6: stop.

(5) construct Binary search tree (BST) from following data: 25, 19, 21, 12, 18, 11, 9, ~~8~~, 30.

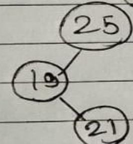
(1) Insert 25.
Let, 25 be the root node



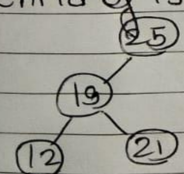
(2) Insert 19. 19<25 Hence 19 becomes the left child of 25



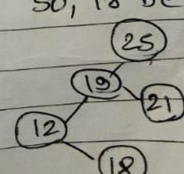
(3) Insert 21. 21<25 and 19<21 so, 21 becomes ~~the~~ right child of 19.



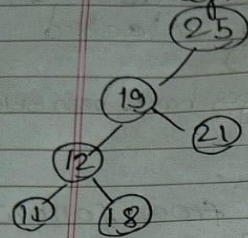
(4) Insert 12. 12<25 and 19>12 so, 12 becomes left child of 19.



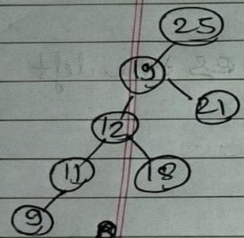
(5) Insert 18 18<25 and 18<19 and 18>12 so, 18 becomes right child of 12



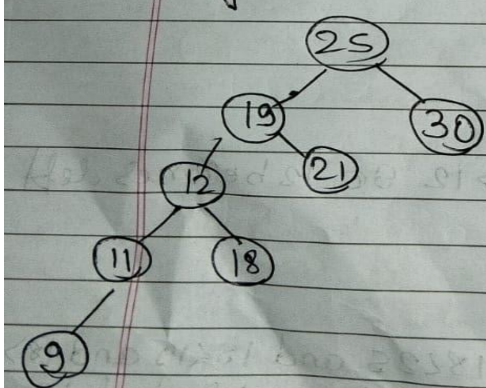
6. Insert 11, $11<25$, $11<19$, $11<12$ so, 11 becomes left child of 12



7. Insert 9, $9<25$, $9<19$, $9<12$, $9<11$ So, 9 becomes the left child of 11



8. Insert 30, $30>25$ so, 30 becomes right child of 25



(6) Consider the set {a,b,c,d}. In how many way can we select two of these letters (repetition is not allowed) when (i) order matters

(ii) order doesnot matter.

Sol^n

(i) If order matters, the number of ways of selecting two letters from four letters is

$$P(4,2) = \frac{4!}{(4-2)!} \qquad \left[\because {}^nP_r = \frac{n!}{(n-r)!}\right]$$

(ii) If order doesnot matter, the number of way of selecting two letters ~~four~~ from four letters is:

$$C(4,2) = \frac{4!}{2!\,(4-2)!} \qquad \left[\because C(n,r) = \frac{n!}{r!\,(n-r)!}\right]$$

$$= \frac{4!}{2!\,2!} = 6 \text{ ways.}$$

(7) Find the GCD of 20 and 28 using Euclidean algorithm.

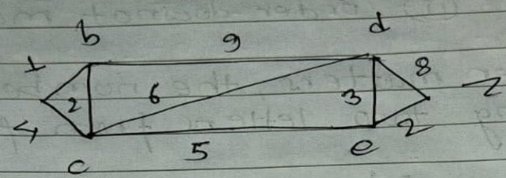
Let, a=28 b=20

q	a	b	r
1	28	20	8
2	20	8	4.
2	8	4	0
	4	0	

when the remainder becomes 0, the divisor at that step is the GCD

So, $\therefore$ GCD (28, 20) = 4

(8) Use Dijkstra's algorithm to find the length of the shortest path between the vertices a and z in the weighted graph displayed below:



Sol:^n Here, $L(a) = 0$

$T = \{b, c, d, e, z\}$

And for all vertices $x \neq a \in T$

$L(x) = \infty$

we have,

$L(x) = \min\{L(x), L(v) + w(v, x)\}$

Iteration: 1

since, $L(a) = 0$ is minimum, $v = a$.

for $v = a$, adjacent vertices $= \{b, c\}$

$L(b) = \min\{L(b), L(a) + w(a, b)\}$

$= \min\{\alpha, 0 + 1) = 1$

$L(c) = \min\{L(c), L(a) + w(a, c)\}$

$= \min\{\infty, 0 + 4\} = 4$

Iteration: 2

since, $L(b) = 1$ is minimum, $v = b$

for $v = b$ adjacent vertices $= \{c, d\}$

$L(c) = \min\{L(c), L(b) + w(b, c)\}$

$= \min\{4, 1 + 2\} = 3$

$$L(d) = \min\{L(d), L(b) + w(b,d)\}$$
$$= \min\{\infty, 1+9\} = 10.$$

Iteration 3

Since, $L(c) = 3$ is minimum, $v = c$

For $v = c$, adjacent vertices $= \{d, e\}$

$$\therefore L(d) = \min\{L(d), L(c) + w(c,d)\}$$
$$= \min\{10, 3+6\} = 9$$
$$L(e) = \min\{L(e), L(c) + w(c,e)\}$$
$$= \min\{\infty, 3+5\} = 8$$

Iteration 4

since, $L(e) = 8$ is minimum, $v = e$

For $v = e$, adjacent vertices $= \{d, z\}$

$$\therefore L(d) = \min\{L(d), L(e) + w(e,d)\}$$
$$= \min\{9, 8+3\} = 9$$
$$L(z) = \min\{L(z), L(e) + w(e,z)\}$$
$$= \min\{\infty, 8+2\} = 10.$$

Iteration 5

Since, $L(d) = 9$ is minimum, $v = d$

for $v = d$, adjacent vertex $= \{z\}$

$$\therefore L(z) = \min\{L(z), L(d) + w(d,d)\}$$
$$= \min\{10, 9+8\} = 10$$

$\therefore$ Length of the shortest path is 10 and the shortest path is $\{a, b, c, e, z\}$.