	COLLEGE OF COMPUTING AND INFORMATION SCIENCES		
	Final Assessment Summer 2021 Semester		
Class Id	107200, 107201	**Course Title**	DAA
Program	BSCS	**Campus / Shift**	Main Campus / Morning
Date	29th July 2021	**Total points**	140
Duration	03 hours	**Faculty Name**	Samrina Zamir, Faisal Ahmed
Student Id	9536	**Student Name**	Muhammad Faraz

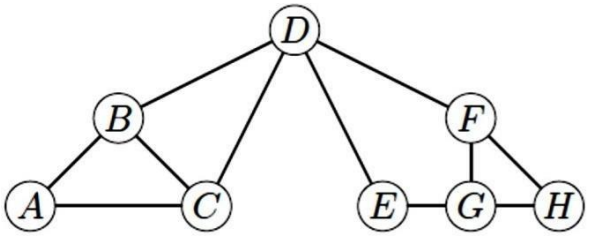
Instructions:

- Filling out Student-ID and Student-Name on exam header is mandatory.
- Do not remove or change any part of exam header or question paper.
- Write down your answers in given space or at the end of exam paper with proper title "Answer for Question# _ _".
- Answers should be formatted correctly (font size, alignment and etc)
- Handwritten text or image should be on A4 size page with clear visibility of contents.
- Only PDF format is accepted (Student are advised to install necessary software)
- In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
- A mandatory recorded viva session will be conducted to ascertain the quality of answer scripts where deemed necessary.

Caution: Duration to perform Final Assessment is **03 hours only**. **Therefore, if you failed to upload answer sheet on LMS (in PDF format) within 03 hours limit, you would be considered as ABSENT/FAILED.**

Question No 1 [5+5+10]

You are playing *Snowstorm's* new video game, *Maze Craft*. Realizing that you can convert a maze into a graph with vertices representing cells and edges representing passages, you want to use your newly learned graph- search algorithms to navigate the maze. Consider the following converted graph.

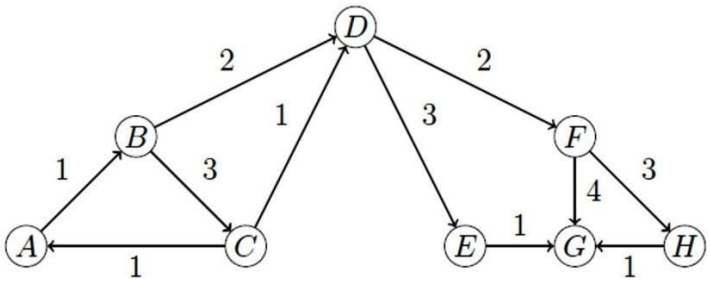


For the following questions, assume that the graph is represented using adjacency lists, and that all adjacency lists are sorted, i.e., the vertices in an adjacency list are always sorted alphabetically.

(i) Suppose that you want to find a path from *A* to *H*. If you use breadth-first search, write down the resulting path as a sequence of vertices.

(ii) If you use depth-first search to find a path from *A* to *H*, write down the resulting path as a sequence of vertices

(iii) Suppose each passage in the maze causes a different amount of damage to you in game. You change the graph to use weights to represent the damage caused by each edge. You then use Dijkstra's algorithm to find the path from *A* to *H* with the lowest possible damage. Write down the order in which vertices get removed from the priority queue when running Dijkstra's algorithm

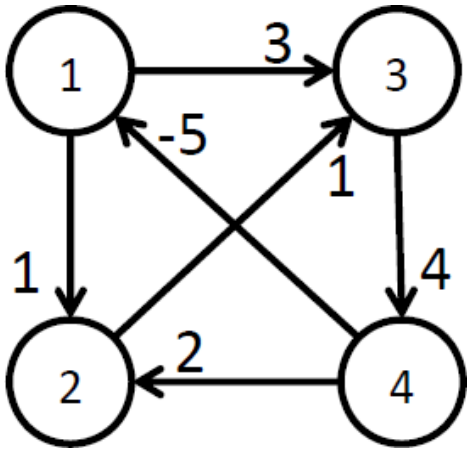


Show working for full credit (calculation of relaxation at each node)

Question No 2 [10+5+5]

- Show the execution of the Belmen-Ford algorithm on the following graph be as descriptive as possible so that the various steps that the algorithm performs are clearly shown.

- Provide the pseudo-code of the Belmen-Ford algorithm.



- Prove that the time Efficiency of Belmen-Ford Algorithm.

Question No 3[7+13]

a) Is the given array a min-heap? Give reasons of your answer and draw the tree.

23	17	14	15	13	10	1	5	7	12

b) Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write checks in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transaction ordering to check-number ordering is therefore the problem of sorting almost-sorted input.

If we have INSERTION-SORT &HEAPSORT programs, which procedure will take less time in this scenario?

Give reason(s) of your answer& apply your selected algorithm on the given array.

23	17	14	15	13	10	1	5	7	12

Question No 4 [10+10]

TCS has hired you to minimize its parcel transportation time from Karachi to every other city,
You must find out all Flight Chains (sequence of flights/ flight routes) from Karachi (KHI) to every other city so that we could deliver mails (parcel) AS QUICKLY AS POSSIBLE. The flight information is available in the table on the right. You have decided to achieve this using the following three steps:

Flight #	From	To	Flight Duration (min)
F1	KHI	HYD	150
F2	KHI	SKR	360
F3	HYD	SKR	200
F4	HYD	ISL	800
F5	ROH	KHI	300
F6	ROH	LHR	860
F7	SKR	ROH	100
F8	SKR	ISL	500
F9	SKR	LHR	450
F10	LHR	KHI	900
F11	SKR	KHI	350

a) Draw the graph. Remember that a flight from KHI to HYD does not mean that there is a flight from HYD to KHI.

b) Apply Dynamic programming technique (algorithm) to find the flight chains. Show all steps.

Question No 5[20]

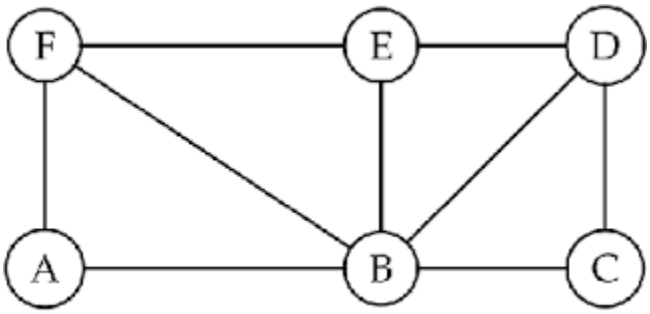
DARAZ.PK has hired you to minimize its parcel transportation time from its Mega store in Area **'C'** to different dispatch centres in Areas marked as; **'A', 'B', 'D, 'E', &'F' in given graph.**

You must find out all rider's routes Mega store(A) to every other dispatch centre so that we could deliver (parcel) AS QUICKLY AS POSSIBLE.

Apply one of the studied algorithms which would find these paths in **O(|v| +|E|)**asymptotic time.

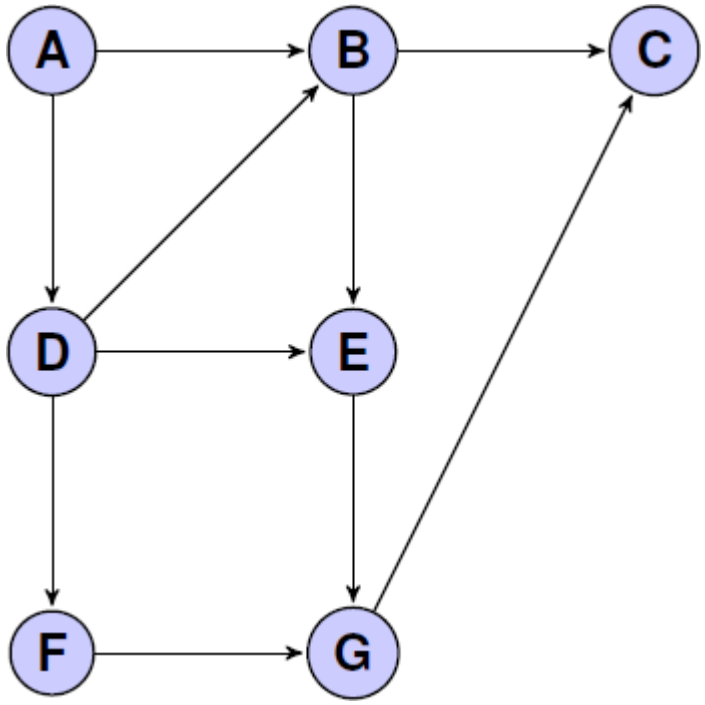
Show all steps,

where, |v| = No. of nodes & |E| = No of Edges.



Question No 6 [15+5]

a) Give a topological sort ordering of the vertices in the following directed acyclic graph (in other words, give a linearized ordering of the vertices in the graph).



b) A *topological sort* of a directed acyclic graph (a graph without cycles) yields a list of vertices such that if there is a path from vertex *i* to vertex *j*, then *i* precedes *j* in the topological sort. In *O*() notation, what is the running time of a topological sort on a graph with |*V*| vertices and |*E*| edges? Give a brief but precise explanation justifying your answer.

Question No 7 [3+5+5+7]

1. What is the total number of nodes in Complete Binary tree of Height 'h'.
2. Given below is the procedure of Max-Heapify

```
MAX-HEAPIFY(A, i)
 1  l = LEFT(i) // 2i
 2  r = RIGHT(i) // 2i + 1
 3  if l ≤ A.heap-size and A[l] > A[i]
 4        largest = l
 5  else
 6        largest = i
 7  if r ≤ A.heap-size and A[r] > A[largest]
 8        largest = r
 9  if i ≠ largest
10        swap(A[i], A[largets])
11        MAX-HEAPIFY(A, largest)
```

- Show that the Recurrence Relation for max-Heapify is

$$T(n) \leq T\left(\frac{2}{3}n\right) + \Theta(1).$$

- Prove By recursion tree or Back substitution Method that the time complexity of Max-Heapify is ***O(log n)***

3. This is the procedure of Build-Heap

```
BUILD-MAX-HEAP(A)
1  A.heap-size = A.length // initialize heap size
2  for i = ⌊A.length/2⌋ downto 1 // 1 is the root of tree
3        MAX-HEAPIFY(A, i)
```

Prove that the time complexity of Build Heap is **not O(n log n)** but ***O(n).***

9536 (1)

:- QUESTION # 01 :-

(i) A, B, C, D, E, F, G, H

(ii) A, B, D, E, G, H

(iii)

Vertex	Priority A Distance from A	Last Vertex
A	0	
B	1	A
C	4	B
D	3	B
E	6	D
F	5	D
G	7	E
H	8	F

According to Priority

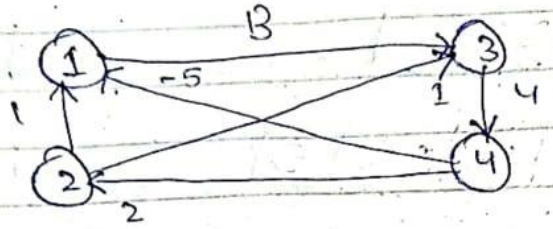
A, B, D, C, F, E, G, H

01536 (2)

:- QUESTION # 02 :-

(a)

Execution of Belmen-ford



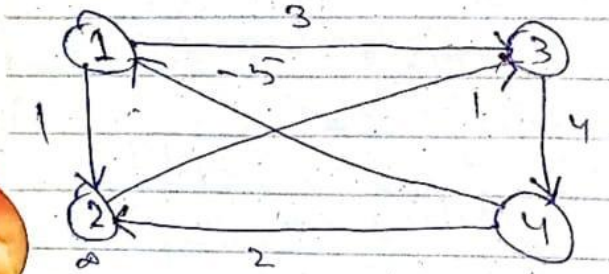
= V-1 ∵ V = Vertices

= 4-1

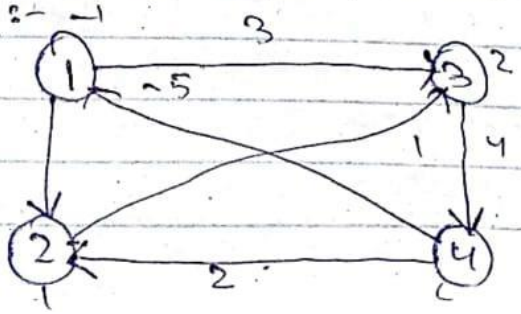
= 3

There will be 3 itteration ($R_1 R_2 R_3$)

R_1 :-

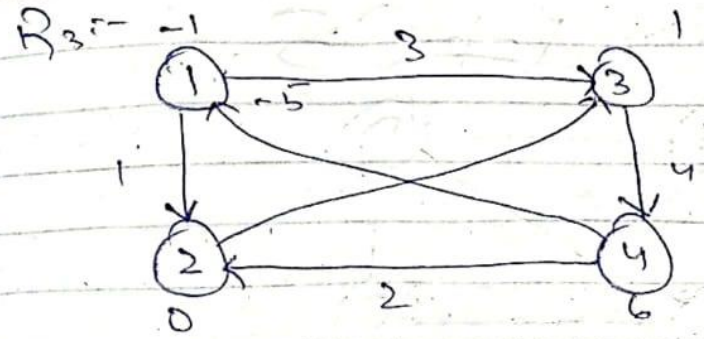


R_2 :-



9536

(3)



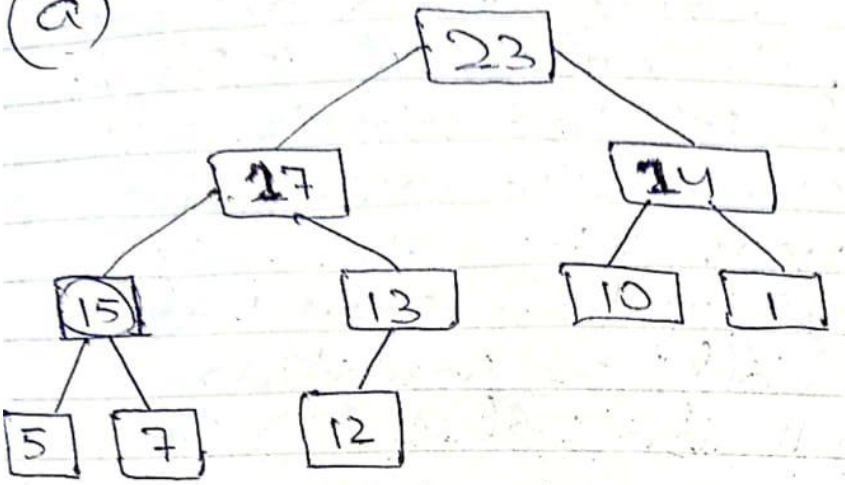
(b) Belmenfood (G, w, s)
Initialize-Single-source (G, s)
For each vertex i = 1 to V[G] - 1 do
  for each edge (U, V) in E[G] do
    Relax (U, V, w)
For each edge (U, V) in E[G] do
  if d[U] + w[U, V) < d[V] then
  return false
return true.

(c)
The initialization in line 1 takes O(V) times for loop of lines 2-4 takes O(E) time and for-loop at line 5-7 takes O(E) time. Thus, the Bellmenford algorithm runs in O(VE) time.

9536 (4)

QUESTION # 03:-

(a)



The given Array is not min-heap because min-heap has the property that for every node i other than the root $A[Parent(i)] \leq A[i]$, whereas we can see that in given tree the value stored in parent node is not less than or equal to the value stored in its child nodes.

9536 (5)

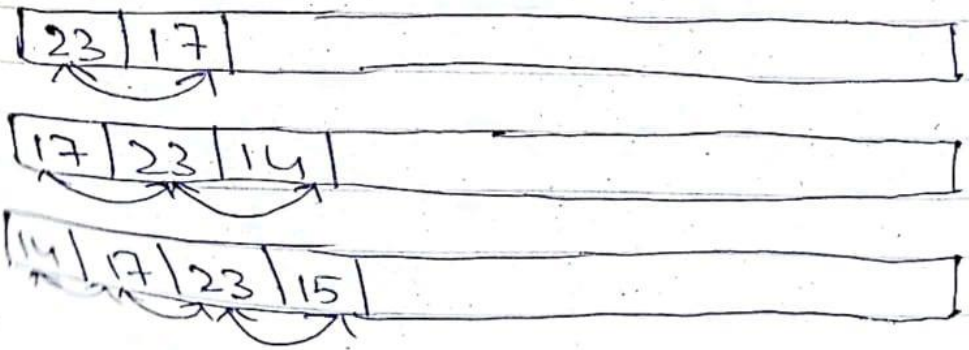
(b)

The more sorted array is, the less work insertion sort will do. Namely sort is O(n+d) where d is the number of inversions in the array. In the example above the number of inversions leads to the small so insertion sort will be close to the linear.

On the other hand, heap sort is always O(nlogn). Each call to HEAPIFY costs O(log(n)) and there are O(n) such calls the running time is thus at most O(n log n).

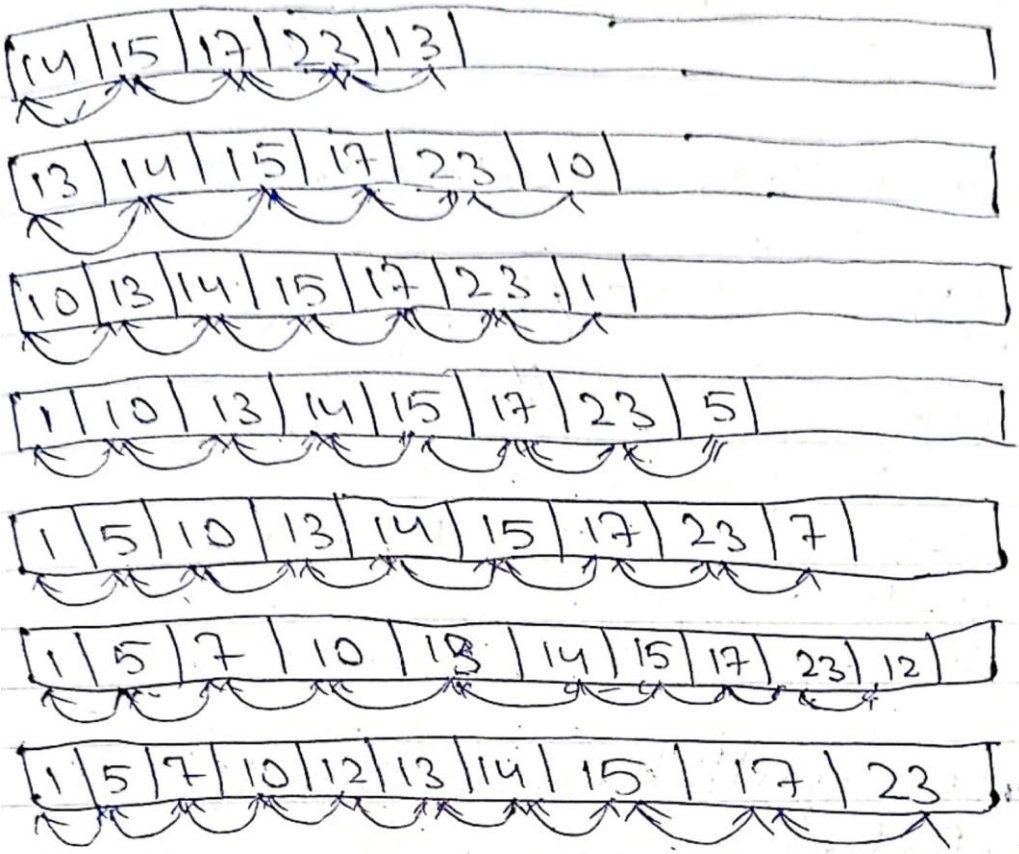
23	17	14	15	13	10	1	5	7	12

Applying Insertion sort



9536

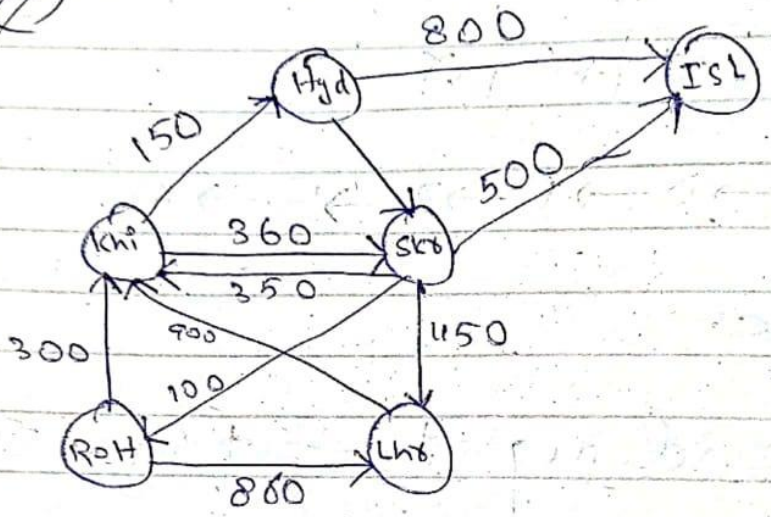
(6)


14 15 17 23 13
13 14 15 17 23 10
10 13 14 15 17 23 1
1 10 13 14 15 17 23 5
1 5 10 13 14 15 17 23 7
1 5 7 10 13 14 15 17 23 12
1 5 7 10 12 13 14 15 17 23

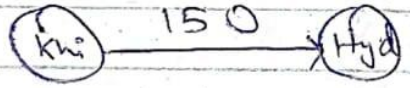
9536 (7)

:- QUESTION # 04 :-

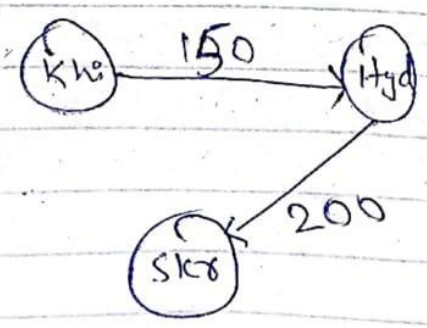
(a)



Karachi to Hyd = 150 :-

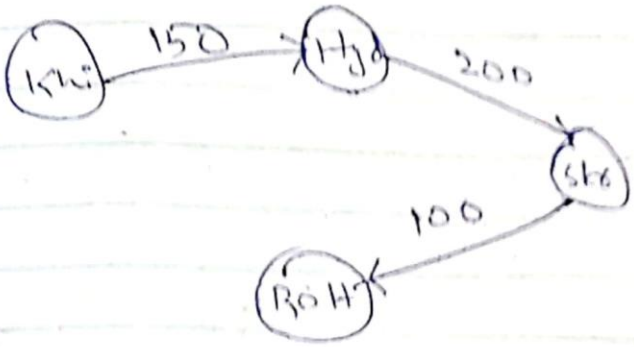


Karachi to Sukkur :- 350 :-

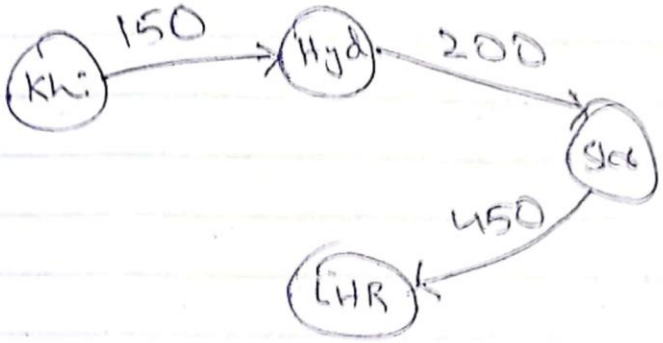


Q536 (R)

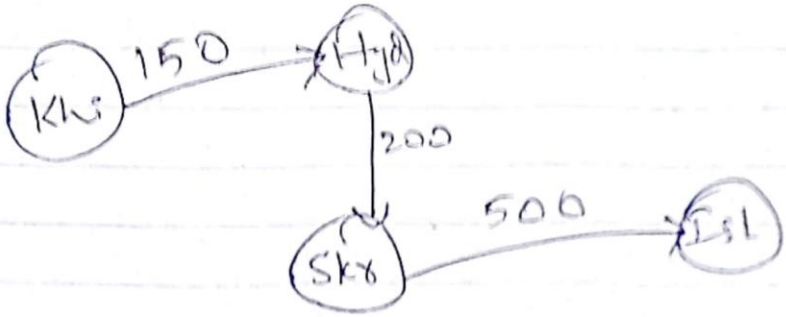
Karachi to ROH = 450 :-



Karachi to LHR :- 800 :-

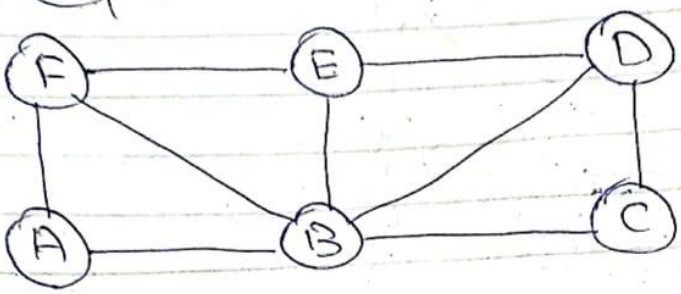


Karachi to ISL :- 850



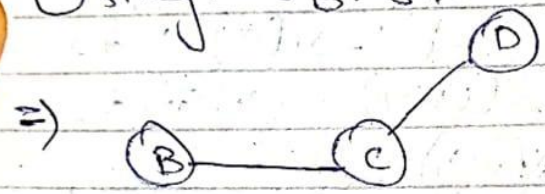
9536 (9)

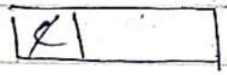
∴ QUESTION # 05 :-

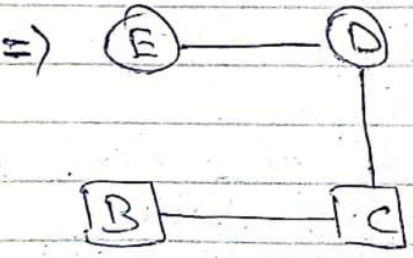


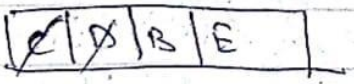
Starting from C :-

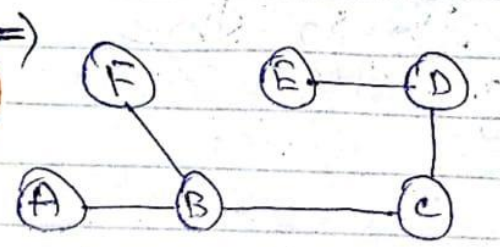
Using BFS :-

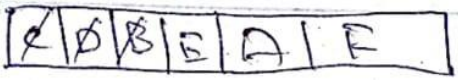




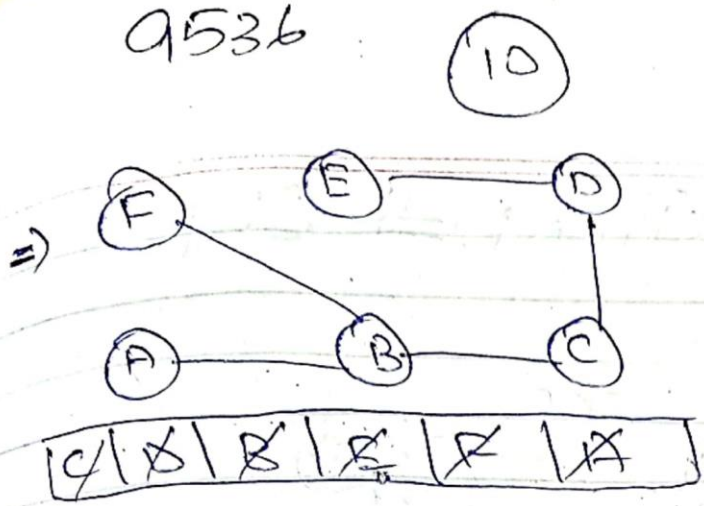








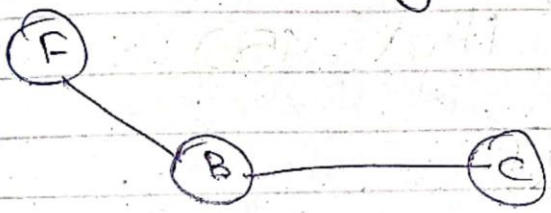
9536



BFS :- C → D → B → E → F → A

Example:-

Parcel sending to F from C

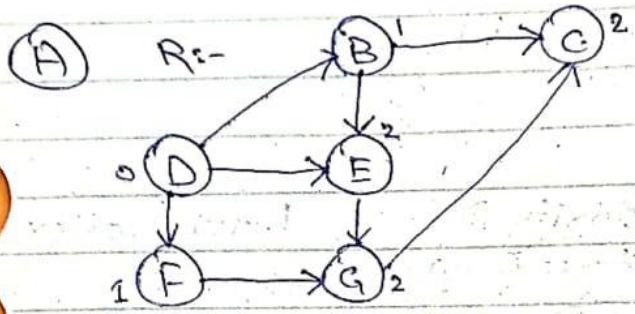


Route C → B — F

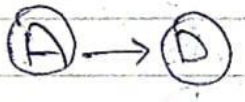
9536 (11)

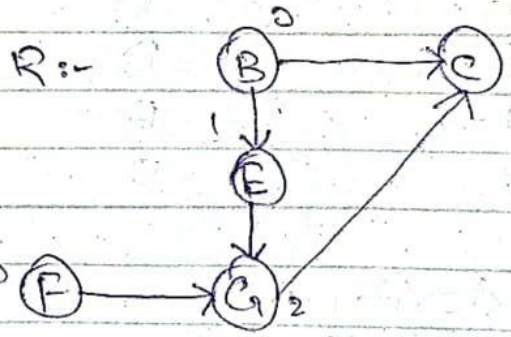
:- QUESTION # 06 :-

(a) Pop (A) Out

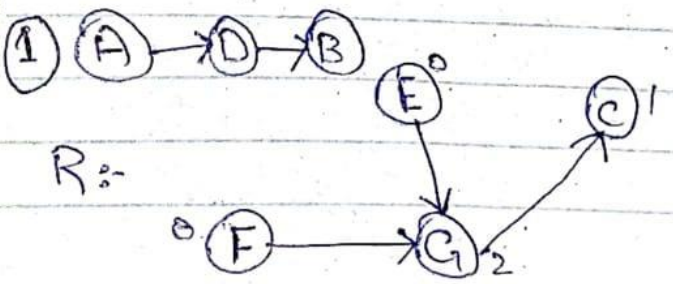


Pop (D) Out:-





Now we can Pop two nodes (2 possibilities)

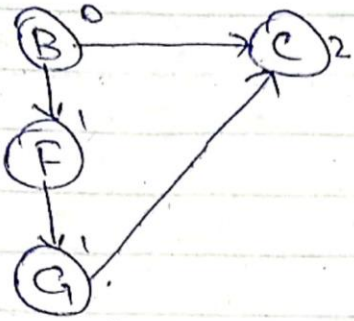


q1536

12

(2) A → D → F

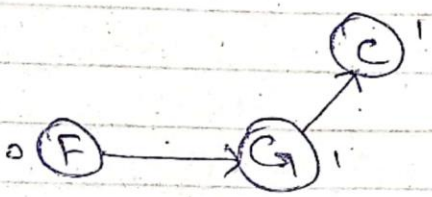
R:-



1) 2 Possible Solutions.

1.1) A → D → B → F

R:-



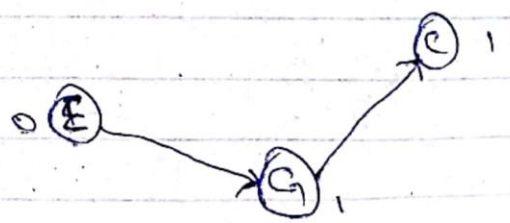
1.2) A → D → B → F

R:-

2) 1 Possible Solution-

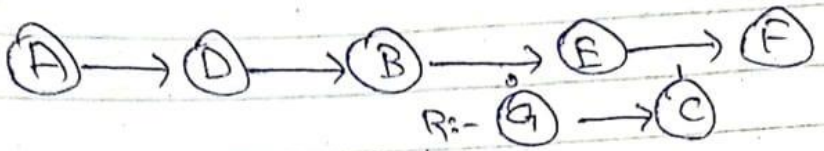
2.1) A → D → F → B

R:-

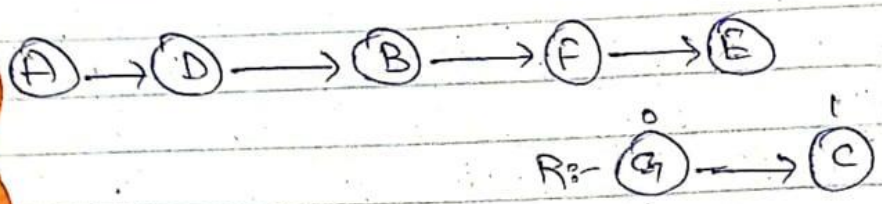


Q1536 (13)

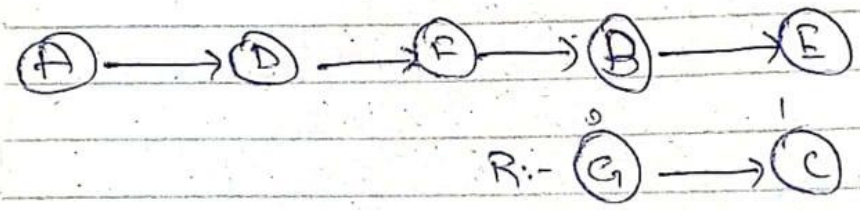
From (1-1) Pop (F) out



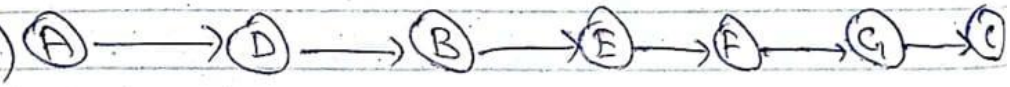
From (1-2) Pop (E) out

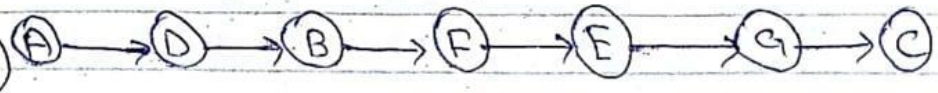


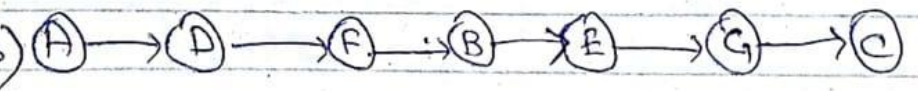
From (2-1) Pop (E) out



All ~~todos~~ topological sort ordering possibilities:-

1) 

2) 

3) 

9536

(14)

(b) Time complexity: The outer for loop will be executed V number of times and the inner for loop will be executed E number of times. Thus overall time capacity is O(E+V).

9536 (15)

QUESTION # 07:-

(1) $2^n - 1$

(2) $n = n_1 + n_2 + 1$

$$\frac{n_1}{n} = \frac{n_1}{n_1 + n_2 + 1}$$

$$= \frac{n_1 / n_2}{n / n_1 + \frac{n_2 + 1}{n_2}}$$

$$= \frac{1}{1 + \frac{n_2 + 1}{n_1}}$$

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

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

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

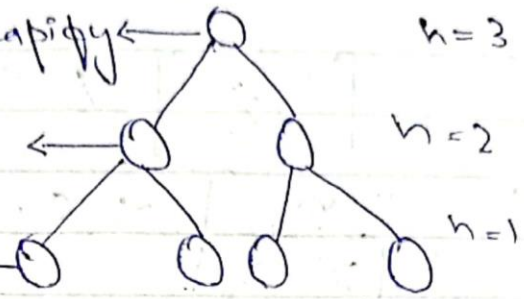
$$= \frac{1}{3/2}$$

Q536 (1.6)

$$\frac{n_L}{n} \leq \frac{2}{3} \Rightarrow n_L = \frac{2}{3}n$$

$$\therefore T(n) = T\left(\tfrac{2}{3}n\right) + \Theta(1)$$

(3) 1 line heapify ← 2 line heapify ← 3 line heapify ←



$$h = 0\left[\frac{15}{2^{h+1}}\right] = \left[\frac{15}{2^{0+1}}\right] = \left[\frac{15}{2}\right] = 8$$

Total time at height h = no of nodes at height "h" log n number of nodes at height $h = \left[\frac{n}{2^{h+1}}\right]$

$$\log n = O(h)$$

total time at height h = no of nodes at height "h" log n

(17)

$$= \left[\frac{n}{2^{n+1}}\right] \times h \cdot c$$

$$= \left[\frac{n}{2^{n-1}}\right] \times O(n)$$

$$= \sum_{n=0}^{\log n} \left[n/2^{n+1}\right] \times c \cdot n$$

$$= \sum_{n=0}^{\log n} \left[\frac{n}{2^n \cdot 2}\right] \times c \cdot n$$

$$= \frac{n \cdot c}{2} \sum_{n=0}^{\log n} \left[\frac{n}{2^n}\right]$$

$$\leq \frac{n \cdot c}{2} \sum_{n=0} \left[\frac{n}{2^n}\right]$$ ← harmonic series sum = 2

$$\leq \frac{n \cdot c}{2} \times 2$$

$$\leq c \cdot n$$

$$\boxed{O(n)}$$