CSE2121: Data Structure

2019 4c

Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true? Explain. 3

i. 9679, 1989, 4199 hashes to the same value
ii. 1471, 6171 has to the same value
iii. All elements hash to the same value
iv. Each element hashes to a different value
v. None of the above.

Now, let's check each statement:

i. 9679, 1989, 4199 hash to the same value
Yes, they all hash to 9. So, this statement is true.

ii. 1471, 6171 has to the same value
Yes, they both hash to 1. So, this statement is also true.

iii. All elements hash to the same value
No, this is not true. The elements hash to different values.

iv. Each element hashes to a different value
No, this is also not true. Some elements hash to the same value.

Therefore, the correct answer is option 'C': i and ii only.

The given input is: 4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199

The hash function is: x mod 10

So, the possible hash values are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Let's apply the hash function to each input:

- 4322 mod 10 = 2
- 1334 mod 10 = 4
- 1471 mod 10 = 1
- 9679 mod 10 = 9
- 1989 mod 10 = 9
- 6171 mod 10 = 1
- 6173 mod 10 = 3
- 4199 mod 10 = 9

2019 5 -c

3(c) Simulate the postfix expression evaluation algorithm using 12, 6, /, 6, 2, +, *, 12, 4, /, − by showing Stack's contents as each element is scanned. (2·75) 2016

element	Stack	Operation
12	12	
6	12, 6	
/	2	[12/6]
6	2, 6	
2	2, 6, 2	
+	2, 8	[6+2]
*	16	[2*8]
12	16, 12	
4	16, 12, 4	
/	16, 3	[12/4]
−	13	[16-3]

2019 6-c

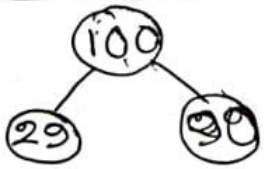
(c) Simulate the maxheap algorithm for the following values: 100, 29, 90, 148, 12, 34, 90, 89, 120, 99 and 12. 3.75

2019

6) (c) Maxheap algorithm-

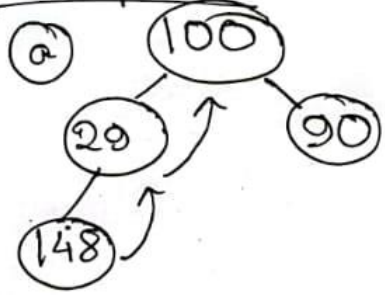
Given values:- 100, 29, 90, 148, 12, 34, 90, 89, 120, 99 & 12

(a) Step-1:

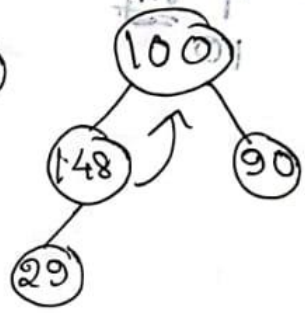


Step-2:

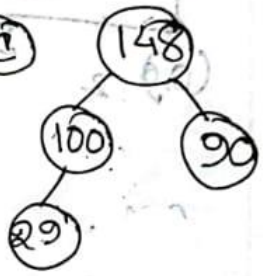
(a)



(b)

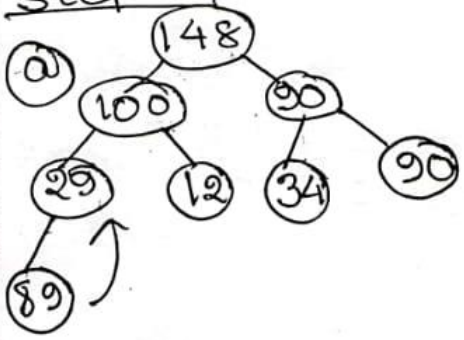


(c)

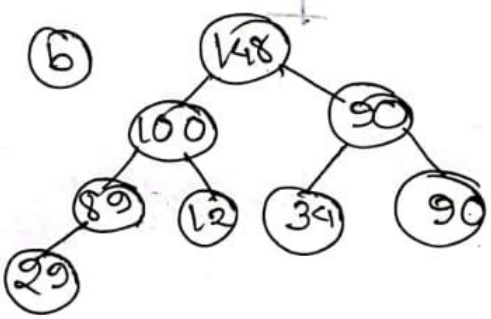


Step-3:

(a)

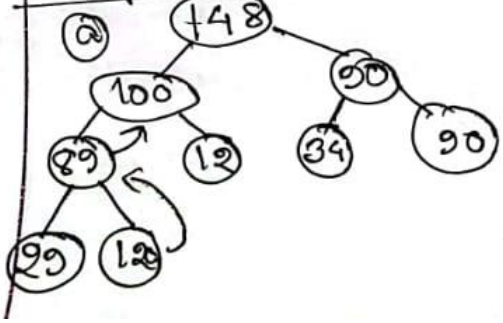


(b)

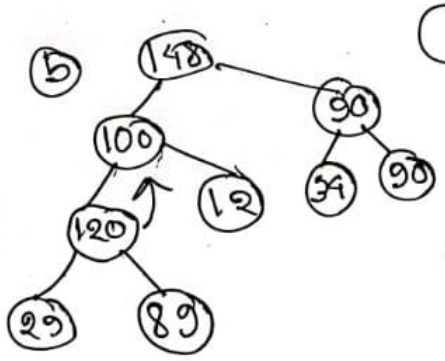


Step-4:-

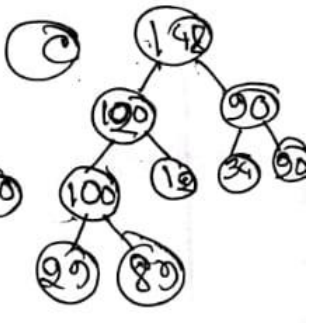
(a)

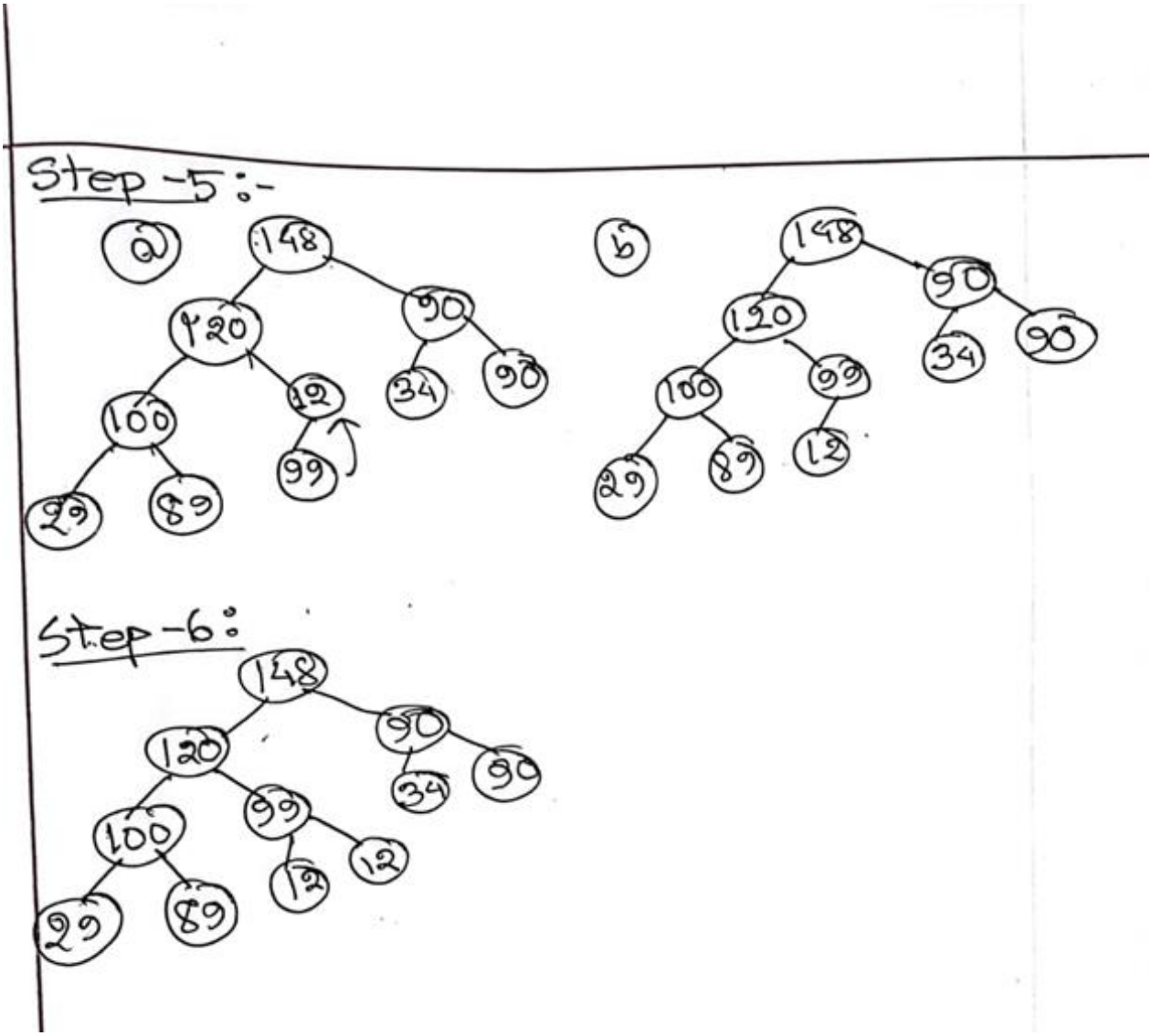


(b)



(c)





2019 7-c

Briefly discuss the advantages of ...

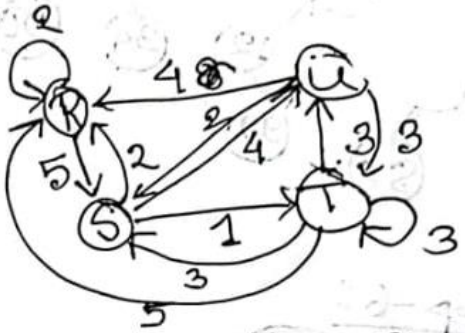
Use the Warshall's algorithm to find the shortest path matrix of the weighted matrix given below. 4.75

$$W = \begin{pmatrix} 2 & 5 & 0 & 0 \\ 2 & 0 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & 0 \end{pmatrix}$$

Warshall's algorithm to find Shortest path matrix of weighted matrix –

$R=V_1, S=V_2, T=V_3, U=V_4$

$$W = \begin{matrix} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{matrix} \begin{matrix} \begin{matrix} V_1 & V_2 & V_3 & V_4 \end{matrix} \\ \begin{pmatrix} 2 & 5 & 0 & 0 \\ 2 & 0 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & 0 \end{pmatrix} \end{matrix}$$



$Q_k[i,j] = MIN(Q_{k-1}[i,j], Q_{k-1}[i,k] + Q_{k-1}[k,j])$

$$Q_0 = \begin{pmatrix} 2 & 5 & \alpha & \alpha \\ 2 & \alpha & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & \alpha \end{pmatrix}$$

$$\begin{pmatrix} RR & RS & RT & - \\ SR & - & ST & SU \\ TR & TS & TT & TU \\ UR & US & UT & UU \end{pmatrix}$$

$$Q_1 = \begin{pmatrix} 2 & 5 & \alpha & \alpha \\ 2 & 7 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & \alpha \end{pmatrix}$$

Q_1 নির্ণয় 1st row & 1st column as it is নিবে।

Min $(D^0[2,2], D^0[2,1] + D^0[1,2])$ এখান 1 দিলে গেলে & direct গেলে যেই path ছোট সেটা নিব।

α , $2+5=7$

$\alpha > 7$. So take 7.

$D^0[2,3], D^0[2,1] + D^0[1,3]$

1 , $2 + \alpha$

1 , α

$1 < \alpha$

$$Q_2 = \begin{pmatrix} 2 & 5 & 6 & 11 \\ 2 & 7 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & 6 \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 2 & 5 & 6 & 6 \\ 2 & 7 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & 6 \end{pmatrix}$$

$$Q_4 = \begin{pmatrix} 2 & 5 & 6 & 6 \\ 2 & 6 & 1 & 4 \\ 5 & 3 & 3 & 3 \\ 4 & 2 & 3 & 6 \end{pmatrix}$$

2019 8c

How many ways a graph G can be ...

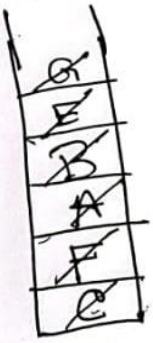
Consider the adjacency list of the Graph G in the following table. Find the nodes that are reachable from node C using Depth First Search. 3

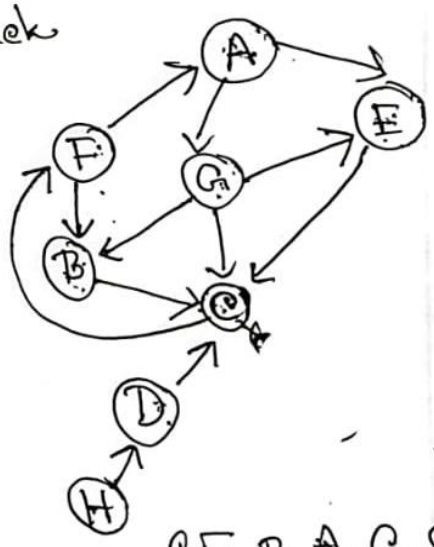
Node	Adjacency	Node	Adjacency
A	G, E	E	C
B	C	F	A, B
C	F	G	B, C, E
D	C	H	D

8 ©:- Reachable from node C using DFS. (3)

DFS traversel: Stack

consider C as root





Result: C, F, B, A, G, E

∴ Reachable from C are - C, F, B, A, G & E

Considering C & to find out all nodes reachable from C (including C itself) use stack.

(a) Initially push C onto stack.
Stack: C.

(b) Pop & Print C, Put F onto stack (neighbour)
Print: C Stack: F

(c) Pop & print F. Push A & B onto stack.
Print: F, Stack: A B.

(d) Pop & print B. There are no nodes in neighborhood which has not been on stack.
Print: B Stack: - A, -

(e) Pop & print A. Push neighbors G & E onto stack
Print: A Stack: G E

(f) Pop E & Print it. No new neighbors.
Print: E Stack: G

(g) Pop & print G. No new neighburs.
Print: G.

∴ Reachable from C →
C, F, B, A, E, G