**CSC017 Fall 2024 Final Exam Sample**

**Student Name： ID：**

|  |  |
| --- | --- |
| **Total Points** |  |

Each multiple choice question has exactly one correct choice as answer. If there are multile choices that are correct, please select the choice “All of the above”. If there are no choice that is correct, please select the choice “None of the above”.

Lec 7. Insert the following keys into the hash tables below, where the hash function is modulo table size (%TableSize). What is the load factor? [1 point] In the top table, resolve collisions with linear probing. In the middle table, resolve collisions with quadratic probing. In the bottom table resolve by separate chaining into a sorted linked list (with the smallest element at the head of the list). If an insertion fails, record which key failed, but attempt to insert any later keys in the list. Do not resize the tables. Insert these keys: 19, 48, 8, 27, 97, 7.

Load factor = number of keys / size of hash table = 6/10 = 0.6

1) with linear probing:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 97 | 7 |  |  |  |  | 27 | 48 | 19 |

2) with quadratic probing: (If the primary hash index is x, probes go to x+1, x+4, x+9, x+16, x+25 and so on, this results in Secondary Clustering.)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  | 97 | 8 |  |  |  | 7 | 27 | 48 | 19 |

3) with separate chaining:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  |  |  |  |  | 27,97, 7 | 48,8 | 19 |

Lec 8. Which of the following statements are true about the following binary search tree? Select all that apply.

21

9

6

20

56

45

15

A. 9 must have been inserted before 56

B. 20 must have been inserted before 6

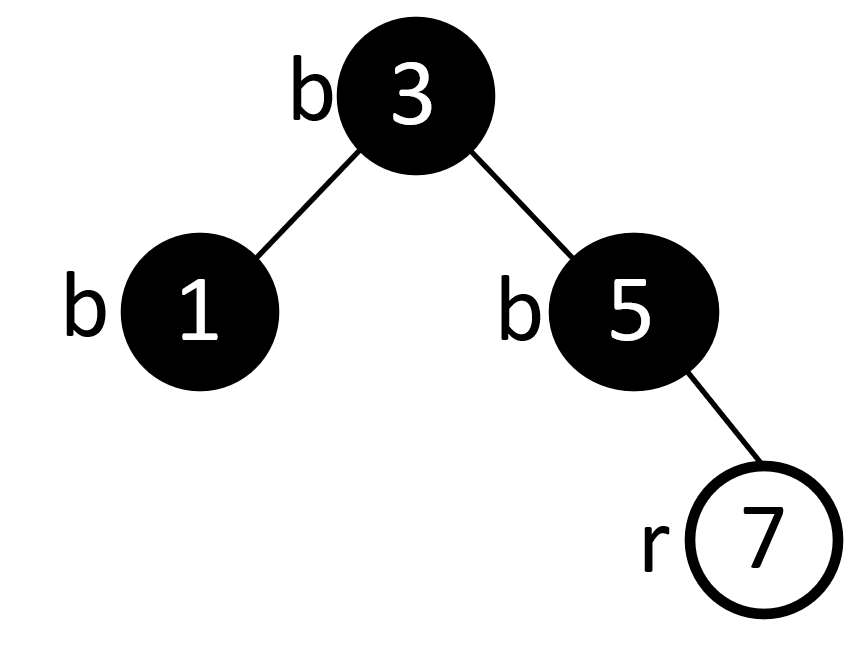
C. 15 must have been inserted last.

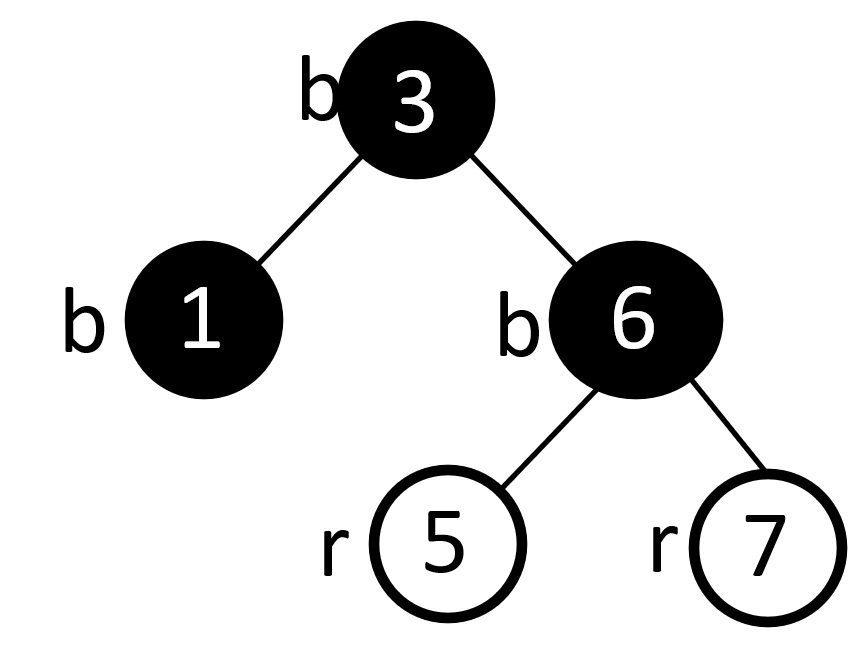
D. 9 must have been inserted before 6

E. All of the above

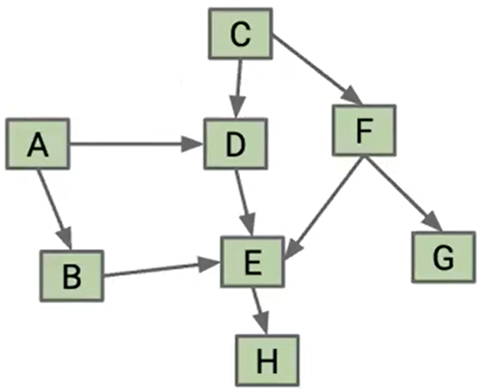
ANS: D

Lec 9. Consider the following red-black tree. Draw the resulting red-black tree after inserting 6. (Due to lack of color printer, I use black filled circles to denote black nodes, and white circiles to denote red nodes. You may put an annotation beside each node, using r for red, b for black)



ANS: 

Lec. 10. Find a topological sort of the following graph, by running depth-first search DFS *starting from node C*, then return vertices in reverse postorder traversal.



ANS: DFS pre-order traversal: C D E H F G A B

DFS post-order traversal: H E D G F C B A

Reverse post-order traversal: A, B, C, F, G, D, E, H

Lec. 10 Which of the following is true about Breadth-First Search (BFS) and Depth-First Search (DFS)?

A. BFS usually finds a shorter path (in terms of the number of nodes) than DFS.

B. DFS uses a Stack to hold the list of nodes that are being explored.

C. BFS uses a Queue to hold the list of unexplored nodes.

D. Both BFS and DFS will always find a path from Start to Goal if there is one.

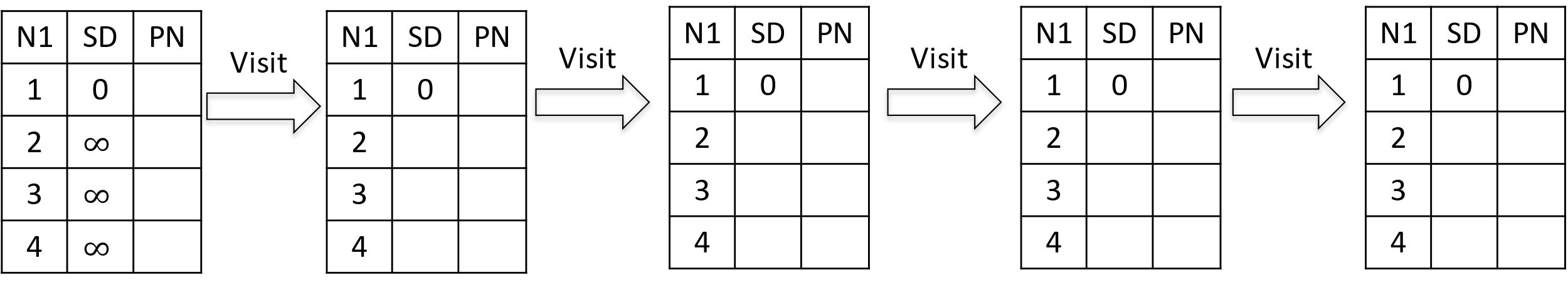
E. All of the above

ANS: E

Lec. 11. Use Dijkstra’s algorithm to find shortest paths starting from source vertex 1 for the following undirected graph. SD: Shortest Distance. PN: Previous Node

A black background with white circles with black numbers

Description automatically generated



ANS:

A black background with a black square

Description automatically generated with medium confidence

Lec. 11. Consider the following weighted digraph. As part of Johnson’s algorithm for All-pairs Shortest Paths, add a dummy source vertex d, and add edges with weight 0 from d to all vertices of G. Let the modified graph be G’.  Compute the shortest distances (instead of running Bellman-Ford algorithm, you can compute it by hand.) h[0], h[1], .. h[V-1]. Then reweight the edges of the original graph to make the edge weights greater than or equal to 0.

A black background with white circles with black numbers

Description automatically generated

ANS: H[1]=0, H[2]=-2, H[3]=0, H[4]=0

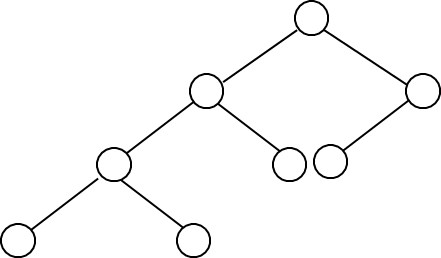
W[1][2]=-2+0-(-2)=0, W[1][3]=1+0-0=1, W[3][2]=1+0-(-2)=3, W[1][4]=1+0-0=1

(You do not have to draw the figures below.)

A number in a circle

Description automatically generated with medium confidence

Lec13. Is this a complete binary tree?



A. Yes

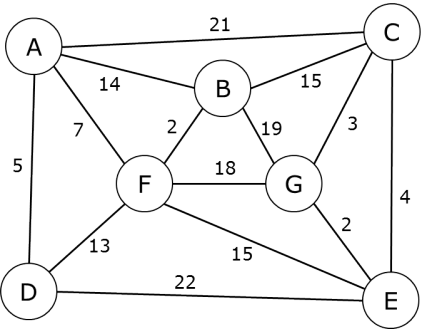
B. No

ANS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

B

Lec 12. Minimum spanning tree.

1) Use Prim’s algorithm to calculate a minimum spanning tree starting from vertex A. If during your algorithm two unvisited vertices have the same distance, use alphabetical order to determine which one is selected first. Name edges in alphabetical order, e.g., write AF instead of FA for the undirected edge. List the edges in the order which Prim’s algorithm includes them into the MST:



ANS:

AD, AF, BF, BC, CG. CE

(You do not need to draw the MST below.)

A diagram of a network

Description automatically generated

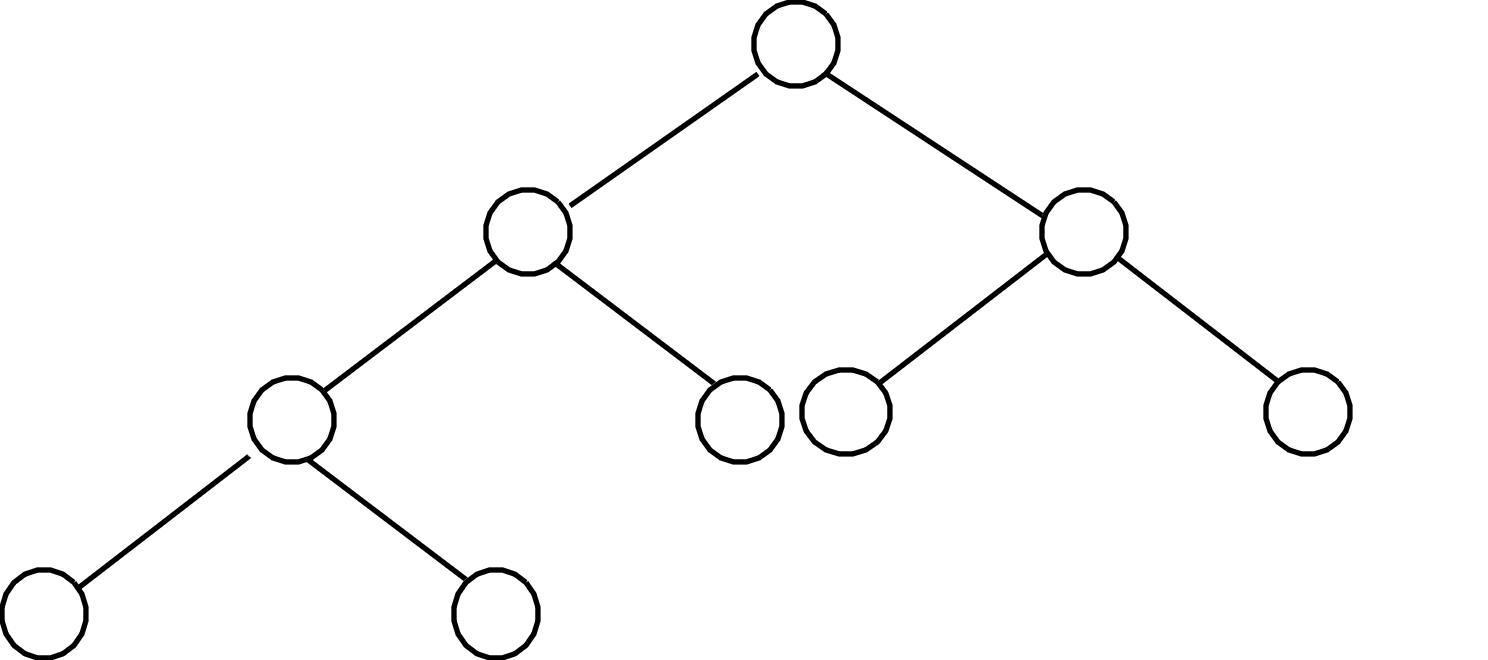
2) List the edges in the order which Kruskal’s algorithm includes them into the MST:

ANS: EG, BF, CG, AD, AF, EF

A diagram of a network

Description automatically generated

Lec13 Is this a complete binary tree?



A. Yes

B. No

ANS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A

Lec 13. The first row below shows a list of numbers to be sorted using mergesort. Show the split into sublists, then show the merge steps. When there are an odd number of elements in a list, make the left sublist larger. Put an ‘X’ on any sublist you don’t use.

65 17 19 85 97 12 23

\_ \_ \_ \_ \_ \_ \_ \_

ANS: A white background with numbers

Description automatically generated

Lec. 13. The first row below shows a list of numbers to be sorted using quicksort. Use *the first number* of each sublist as the pivot. In the second row of the chart, enter the pivot in the circle. Then enter the numbers in the left and right sublists to the left and right of the pivot, respectively. (Note that the two sublists need not be the same size.) Repeat this process on every line. If a sublist is empty, just draw an X on it. Draw the coresponding Binary Search Tree. Give the final sorted list.

**65 17 19 85 97 12 23**

ANS:

**65 17 19 85 97 12 23**

65

65

**17 19 12 23 85 97**

12 **19 23 X 97**

17

85

85

17

**X**

12

**X**

**X** 19

**23**

**X**

X

**X**

**X** 97

**X**

65

17

12

19

23

85

97

Sorted list: 12 17 19 23 65 85 97

Lec 14. Perform a radix sort of the following list of numbers, using a radix of 10, into ascending order. Show the intermediate results after each pass.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 130 | 54 | 57 | 9 | 208 | 42 | 2 | 66 |

After 1st pass

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

After 2nd pass

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

After 3rd pass

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

ANS:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 130 | 54 | 57 | 9 | 208 | 42 | 2 | 66 |

After 1st pass (sorting by the last digit)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 130 | 42 | 2 | 54 | 57 | 66 | 208 | 9 |

After 2nd pass (sorting by the 2nd to last digit)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 208 | 9 | 130 | 42 | 54 | 57 | 66 |

After 3rd pass (sorting by the first digit)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 9 | 42 | 54 | 57 | 66 | 130 | 208 |