Data Structure, Final exam

A. Single Choice Questions (5% each, 60%)

(1) Which of the following statements is TRUE?

(A) The time complexity of the binary search algorithm is $O(n \log n)$.

(B) In a map ADT, when we call the `put()` method and if the key is already in the map, it replaces the old value with the new value.

(C) Both the bubble sort and the selection sort require n passes to sort n items.

(D) The time complexity of quick sort is consistently $O(n \log n)$, independent of the array's initial arrangement.

(B)

(2) Given the following list of numbers: `[5, 16, 20, 12, 3, 8, 9, 17, 19, 7, 11]`. Which answer illustrates the contents of the list after all swapping is complete for a gap size of 4 in a shell sort?

(A) `[3, 7, 9, 12, 5, 8, 11, 17, 19, 16, 20]`

(B) `[3, 7, 5, 8, 9, 12, 11, 16, 19, 17, 20]`

(C) `[5, 9, 17, 12, 3, 8, 16, 20, 19, 7, 11]`

(D) `[3, 5, 7, 8, 9, 11, 12, 16, 17, 19, 20]`

(A)

(3) What creates hash collisions?

(A) When pairs of different hash values are mapped to the same key.

(B) When pairs of different hash values do not share the same key.

(C) When pairs of different keys do not share the same hash value.

(D) When pairs of different keys are mapped to the same hash value.

(D)

(4) Which of the following lists could not be obtained at any point when applying the quicksort algorithm to the list below, assuming the first element is always selected as the pivot value?

```
[4, 2, 3, 7, 9, 11, 5]
```

(A) `[3, 2, 4, 7, 9, 11, 5]`

(B) `[2, 3, 4, 5, 7, 9, 11]`

(C) `[2, 3, 4, 5, 7, 11, 9]`

(D) `[3, 2, 4, 7, 5, 9, 11]`

(D)

(5) What is the total number of nodes you will visit when you search for the number 28 in the following binary search tree?

```
     17
    /  \
    6   19
  / \ / \
  3 9 18 22
```

(A) 2

(B) 3

(C) 4

(D) 5

(B)

(6) Consider the list `[4, 14, 5, 21, 29, 12, 16]`. Identify the FALSE statement:

(A) `[4]` and `[14]` are the first two lists to be merged when using merge sort.

(B) 16 will be selected as the pivot value when using the median of three in quicksort.

(C) After sorting the list in ascending order using bubble sort, a sequential search will find the number 14 more quickly than a binary search would.

(D) Insertion sort will need 6 passes to sort the list.

(C)

(7) Inserting the following keys `6, 15, 3, 9, 4, 10, 8` in order into a binary search tree. Which of the following is FALSE?

(A) The preorder traversal of the tree will be `4, 3, 8, 10, 9, 15, 6`

(B) The inorder traversal of the tree will be `3, 4, 6, 8, 9, 10, 15`

(C) Finding the node with 8 will go through three edges

(D) The nodes 4 and 9 are at the same level

(A)

(8) Which of the following statements is TRUE?

(A) In the knight's tour problem, Warnsdorf's algorithm prefers to select the node with the most available moves.

(B) The DFS algorithm is commonly employed to solve the word ladder problem.

(C) In Dijkstra's algorithm, each node is visited exactly once.

(D) The BFS algorithm can be further divided into preorder, inorder, and postorder traversals in a tree.

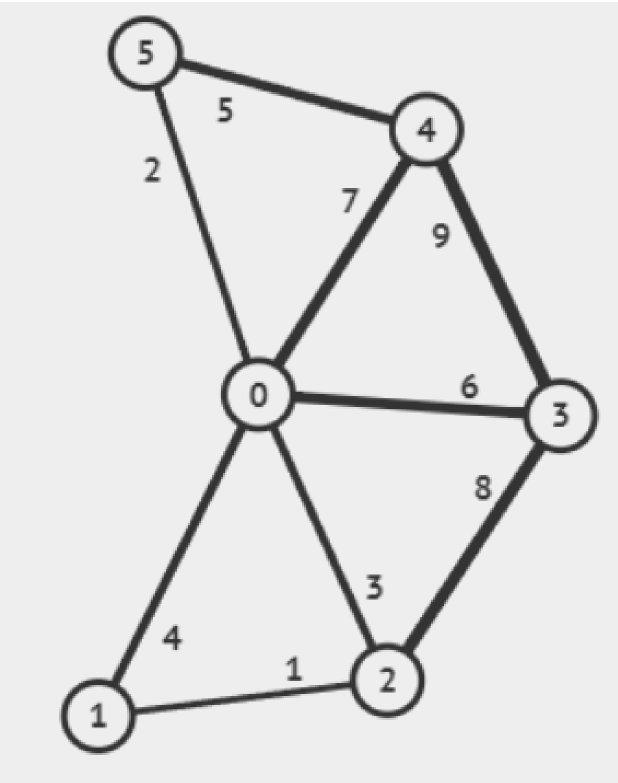
(C)

(9) Which of the following trees is NOT a binary search tree, assuming each number is a node?

```
    (A)                 (B)                 (C)           (D)
       14                  13               13               19
      /  \                /  \                \              /
     6   19             11    19               17           8
    / \                / \                    /            /
   3 8               3    10                14           1
```

(B)

(10) Considering the graph provided, what is the last edge added to the minimum spanning tree when using Prim's algorithm starting from node 0?



(A) `(5, 4)`

(B) `(0, 3)`

(C) `(2, 1)`

(D) `(0, 4)`

(B)

(11) Suppose you have the following list of numbers to sort: `[11, 7, 12, 14, 19, 1, 6, 18, 8, 20]`. Which list represents the partially sorted list after four complete passes of selection sort, assuming that we swap the smallest element to the beginning in each pass?

(A) `[1, 7, 12, 14, 19, 11, 6, 18, 8, 20]`

(B) `[1, 6, 7, 8, 19, 11, 12, 18, 14, 20]`

(C) `[1, 6, 7, 14, 19, 1, 6, 18, 8, 20]`

(D) `[11, 7, 12, 14, 8, 1, 6, 18, 19, 20]`

(B)

(12) Which of the following statements is FALSE?

(A) A queue is usually used to hold nodes in breadth-first search of a graph.

(B) The time complexity for insertion and deletion in a min heap is $O(\log n)$.

(C) The left child of a node with index i can be accessed using $2 \times i + 2$ in a complete binary tree.

(D) The time complexity of DFS and BFS for a tree can be reduced to $O(|V|)$.

(C)

B. Short-answer questions, Please provide the derivation for each question along with your answer (8% each, 40%).

(13) Briefly explain each of the following terms.

(a) Binary Search

(b) Hash Table

(c) Depth-First Search (DFS)

(d) Min-heap

(a) **Binary Search**: A search algorithm applied on a sorted array that repeatedly divides the search range in half, comparing the midpoint against the target number to find it efficiently.

(b) **Hash Table**: A data structure that implements an associative array abstract data type, allowing for the efficient retrieval of values keyed by numbers, through hashing.

(c) **Depth-First Search (DFS)**: An algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node and explores as far as possible along each branch before backtracking.

(d) **Min-heap** : A min-heap is a specialized tree-based data structure in which each node's value is less than or equal to the values of its children. This arrangement ensures that the smallest value is always at the root, which is beneficial for applications needing quick access to the smallest element, such as implementing efficient priority queues and Dijkstra's shortest path algorithm.

(14) Suppose you are given the following set of keys to insert into a hash table that holds exactly 13 values: `25, 117, 97, 100, 114, 108, 116, 105, 99, 17`. What are the contents of the hash table after all the keys have been inserted using the remainder method and linear probing?

Here are the steps, calculating the hash for each key and placing it into the table, resolving collisions using linear probing:

1. **Hash Table Setup**: The hash table has 13 slots, initially empty:

   ```
   [ -, -, -, -, -, -, -, -, -, -, -, -, - ]
   ```

2. **Inserting Each Key**:

   - **Key 25**: $25 \mod 13 = 12$

`[ -, -, -, -, -, -, -, -, -, -, -, -, 25 ]`

- **Key 117**: $117 \mod 13 = 0$ `[ 117, -, -, -, -, -, -, -, -, -, -, -, 25 ]`
- **Key 97**: $97 \mod 13 = 6$

  `[ 117, -, -, -, -, -, 97, -, -, -, -, -, 25 ]`
- **Key 100**: $100 \mod 13 = 9$

  `[ 117, -, -, -, -, -, 97, -, -, 100, -, -, 25 ]`
- **Key 114**: $114 \mod 13 = 10$

  `[ 117, -, -, -, -, -, 97, -, -, 100, 114, -, 25 ]`
- **Key 108**: $108 \mod 13 = 4$

  `[ 117, -, -, -, 108, -, 97, -, -, 100, 114, -, 25 ]`
- **Key 116**: $116 \mod 13 = 12$, but slot 12 is occupied, so linear probe to slot 0, then to slot 1.

  `[ 117, 116, -, -, 108, -, 97, -, -, 100, 114, -, 25 ]`
- **Key 105**: $105 \mod 13 = 1$, but slot 1 is occupied, so linear probe to slot 2.

  `[ 117, 116, 105, -, 108, -, 97, -, -, 100, 114, -, 25 ]`
- **Key 99**: $99 \mod 13 = 8$

  `[ 117, 116, 105, -, 108, -, 97, -, 99, 100, 114, -, 25 ]`
- **Key 17**: $17 \mod 13 = 4$, but slot 4 is occupied, so linear probe to slot 5.

  `[ 117, 116, 105, -, 108, 17, 97, -, 99, 100, 114, -, 25 ]`

3. **Final Hash Table Content**:

   `[ 117, 116, 105, -, 108, 17, 97, -, 99, 100, 114, -, 25 ]`

Here, `-` indicates an empty slot.

(15) The following tree is the result after inserting `10, 4, 9, 8, 12, 15, 3, 5, 14, 18` into a min heap. What is the result after deleting the root? Draw each step of the delete operation.

```
        3
      /   \
     5     4
    / \   / \
   8  12 15  9
  / \  /
 10 14 18
```

Step 1: After delete 3, we will move 18 to the root

```
       18
      /   \
     5     4
    / \   / \
   8  12 15  9
  / \
 10 14
```

Step 2: We swap 18 and 4 when percolate down

```
         4
       /   \
      5     18
     / \    / \
    8   12 15   9
   / \
  10 14
```

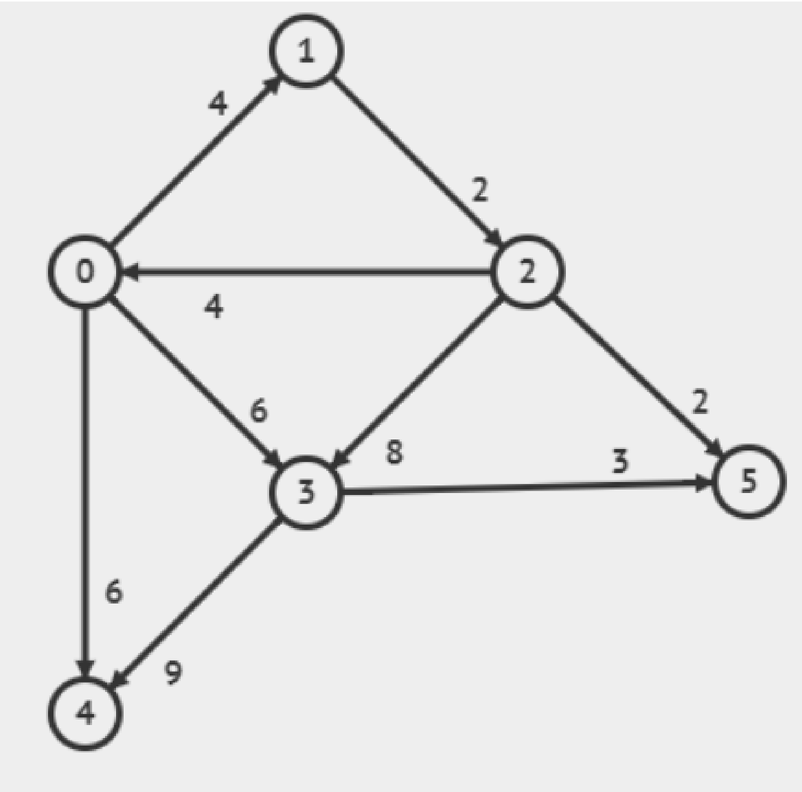
Step 3: We swap 18 and 9 when percolate down

```
         4
       /   \
      5     9
     / \    / \
    8   12 15   18
   / \
  10 14
```

The resulting tree is

```
         4
       /   \
      5     9
     / \    / \
    8   12 15   18
   / \
  10 14
```

(16) Given a weighted graph as follows, answering the following questions:



Using Dijkstra's algorithm, show the order in which the nodes are visited, starting from node 0 and visiting the entire graph. Find the shortest path from node 0 to node 5. When multiple nodes have the same distance, follow the numerical order of the nodes.

1. **Initialization**:

- Distances: $\{0:0, 1:\infty, 2:\infty, 3:\infty, 4:\infty, 5:\infty\}$
- Priority Queue: $[(0,0)]$

2. **First Iteration**:

   - Visit node 0:
     - Neighbors: 1 (distance 4), 3 (distance 6), 4 (distance 6)
     - Updated distances: $\{0:0, 1:4, 2:\infty, 3:6, 4:6, 5:\infty\}$
     - Priority Queue: $[(4,1), (6,3), (6,4)]$

3. **Second Iteration**:

   - Visit node 1 (minimum distance 4):
     - Neighbors: 2 (distance 6)
     - Updated distances: $\{0:0, 1:4, 2:6, 3:6, 4:6, 5:\infty\}$
     - Priority Queue: $[(6,2), (6,3), (6,4)]$

4. **Third Iteration**:

   - Visit node 2 (minimum distance 6):
     - Neighbors: 3 (distance 14), 5 (distance 8)
     - Updated distances: $\{0:0, 1:4, 2:6, 3:6, 4:6, 5:8\}$
     - Priority Queue: $[(6,3), (6,4), (8,5)]$

5. **Fourth Iteration**:

   - Visit node 3 (minimum distance 6):
     - Neighbors: 4 (distance 15), 5 (distance 9)
     - Updated distances: $\{0:0, 1:4, 2:6, 3:6, 4:6, 5:8\}$
     - Priority Queue: $[(6,4), (8,5)]$

6. **Fifth Iteration**:

   - Visit node 4 (minimum distance 6):
     - Neighbors: None
     - Updated distances: $\{0:0, 1:4, 2:6, 3:6, 4:6, 5:8\}$
     - Priority Queue: $[(8,5)]$

7. **Sixth Iteration**:

   - Visit node 5 (minimum distance 8):
     - Final distances: $\{0:0, 1:4, 2:6, 3:6, 4:6, 5:8\}$

Order of Nodes Visited:

The nodes are visited in the following order: 0, 1, 2, 3, 4, 5.

The shortest path from node 0 to node 5 is through the nodes 0 → 1 → 2 → 5 with a total distance of 8.

(17): A binary tree has six nodes. The following are the inorder and postorder traversals of the tree. Can you draw the tree? If yes, please draw it. If the tree does not exist, please explain why not.

```
Postorder: ECDFBA
Inorder: ECADBF
```

Yes.

The postorder traversal ECDFBA tells us that node A is the root. The Inorder traversal ECADBF implies that nodes EC in the left of A are in the left subtree and nodes DBF in the right of A are in the right subtree. Following the same logic for each subtree we build the binary tree.

Step1:

```
          A
        /   \
      EC     DBF
```

Step2:

```
          A
        /   \
      C       B
     /       / \
    E       D   F
```