Data Structure, Quiz 2

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

(1) Which of the following statements about sorting algorithms is TRUE?

(A) Insertion sort performs consistently regardless of the initial order of the array because it always identifies the smallest (or largest) element to position it correctly.

(B) The worst-case scenario for selection sort occurs when the array is in reverse order, requiring each element to be compared and potentially swapped multiple times to reach its correct position.

(C) Merge sort is considered a stable sorting algorithm because, during the merging process, if two elements from the merged arrays are equal, the element from the left (or first) array is always prioritized.

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

(C)

(2) Given the list of numbers `[15, 19, 11, 5, 2, 29, 12, 14, 13, 22]`, identify the resulting array after the first partitioning step in a quicksort algorithm, assuming the first element is used as the pivot. Which of the following represents the correct order?

(A) `[12, 13, 11, 5,  2, 14, 15, 29, 19, 22]`

(B) `[14, 11, 19, 5,  2, 15, 12, 13, 29, 22]`

(C) `[13, 12, 11, 2,  5, 14, 15, 19, 29, 22]`

(D) `[15, 12, 13, 11, 5, 22, 29, 14,  2, 19]`

(A)

(3) Identify the preferred data structures used in BFS (Breadth-First Search), DFS (Depth-First Search), Dijkstra's algorithm, and Prim's algorithm, respectively:

(A) Stack, Queue, Priority Queue, Priority Queue.

(B) Priority Queue, Stack, Priority Queue, Queue.

(C) Queue, Stack, Priority Queue, Hash table.

(D) Queue, Stack, Priority Queue, Priority Queue.

(D)

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

(A)

```
     15
    /
   6
  /
 3
```

(B)

```
     13
```

(C)

```
     13
    / | \
   8  5  3
```

(D)

```
     21
    /   \
   8   23
  /
 1
```

(C)

(5) What is the last node you will visit when you search number 8 in the following binary search tree?

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

(A) 17

(B) 3

(C) 9

(D) 18

(C)

(6) Consider the list `[1,15,5,21,29,12,16]`. Identify the FALSE statement:

(A) Insertion sort requires six passes to completely sort the list.

(B) Using a sequential search, the number 5 will be found more quickly than the number 29.

(C) After sorting the list in ascending order using bubble sort and employing a binary search, the number 15 will be found more quickly than the number 16.

(D) If the list is sorted in ascending order using shell sort, then using a sequential search will find the number 15 more quickly than a binary search would.

(D)

(7) For the postorder traversal sequence of the binary tree below, which of the statement is TRUE?

```
     A
   /   \
  B     D
 /     / \
 E    H   J
/\   /\
K L  F M
```

(A) Node B appears at the second position.

(B) Node M appears after node D.

(C) Node A appears between node B and node D.

(D) Node H appears immediately before node J.

(D)

(8) Which of the following statements is TRUE?

(A) Prim's algorithm identifies the same minimum spanning tree in a graph, regardless of the starting vertex.

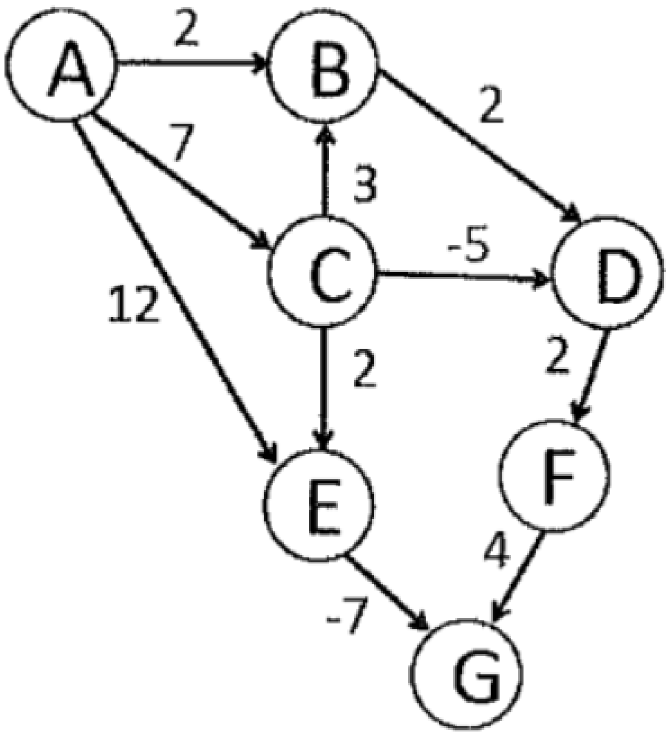
(B) Dijkstra's algorithm cannot find the shortest path correctly if there are negative weights.

(C) The adjacency matrix representation is more memory efficient compared to the adjacency list for a graph.

(D) The BFS algorithm can find the shortest path between two nodes in a weighted graph.

(B)

(9) Given a weighted directed graph with negative edges as follows, what are the vertices in the order they are selcted bt the Dijkstra's algorithm?



(A) `A, B, D, F, C, G, E`

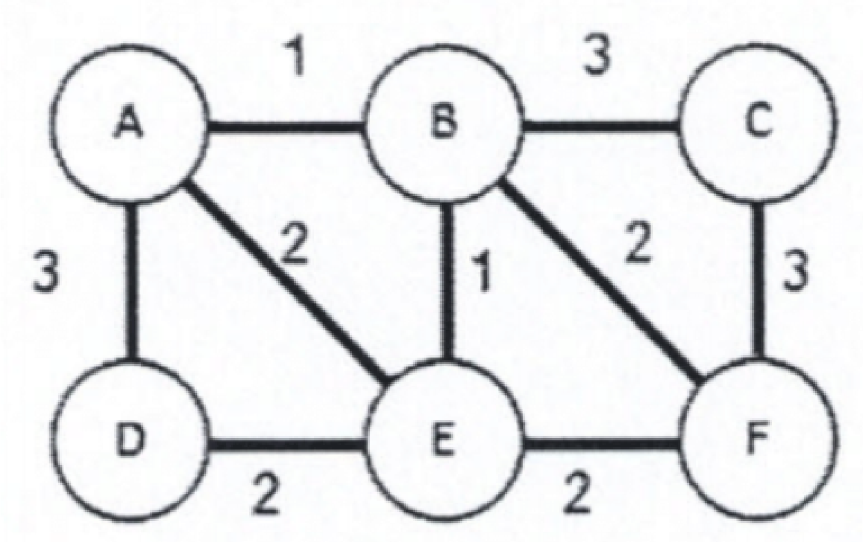
(B) `A, B, D, F, C, E, G`

(C) `A, B, C, F, D, F, G`

(D) `A, B, D, F, G, C, E`

(B)

(10) Considering the graph provided, which edge does not appear in any minimum spanning tree of the graph?



(A) `(B, E)`

(B) `(A, E)`

(C) `(C, F)`

(D) `(B, C)`

(B)

(11) Which of the following lists would NOT be obtained at some point when applying the insertion sort algorithm to the list below?

`[9,2,5,5,7,9,1]`

(A) `[2,9,5,5,7,9,1]`

(B) `[2,5,5,9,7,9,1]`

(C) `[2,5,5,7,9,9,1]`

(D) `[2,5,5,9,9,7,1]`

(D)

(12) Which of the following statements is FALSE?

(A) A binary tree can still be considered a binary tree even if it contains only one element.

(B) In inorder traversal, the left subtree is processed first, followed by the root node, and finally the right subtree.

(C) The balance factor of an AVL tree is always -1, 0, or 1.

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

(Give away questions, all statement are TRUE)

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) Collision Resolution

(b) Stable sort

(c) Priority Queue

(d) Complete Binary Tree

(a) **Collision Resolution**: In the context of hash tables, collision resolution is a set of strategies used to handle the situation where two or more keys hash to the same index in the array. When a collision occurs, the hash table must have a method to resolve the collision and find an alternative location to store the value associated with the key. Common collision resolution techniques include linear probing, quadratic probing and chaining.

(b) **Stable Sort**: A sorting algorithm is stable if it preserves the relative order of records with equal keys. In other words, if two items are equal according to the sort key, their order in the sorted output will be the same as their order in the input. Examples of stable sorting algorithms include merge sort, insertion sort, and bubble sort.

(c) **Priority Queue**: A priority queue is an abstract data type that operates similar to a regular queue except that each element has a certain priority. Elements are added into the queue as they arrive, but the element with the highest priority is always removed first. This differs from a typical queue, where the order of elements is strictly based on their arrival sequence (FIFO order).

(d) **Complete Binary Tree**: A complete binary tree is a type of binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. This structural property is very useful in the efficient implementation of binary heaps. In a complete binary tree, if the nodes are numbered from top to bottom and from left to right, any node with an index i will have its children at indices $2i + 1$ and $2i + 2$ and its parent at index $(i - 1)/2$, assuming indexing starts at 0.

(14) Suppose you are given the following set of keys to insert into a hash table that holds exactly 11 values: `113 , 117 , 97 , 100 , 114 , 108 , 116 , 105 , 99`. What is contents of the hash table after all the keys have been inserted using remainder method and quadratic probing?

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

1. **Hash Table Setup**: The hash table has 11 slots, initially empty:

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

2. **Inserting Each Key**:

   - **Key 113**: $113 \mod 11 = 3$

     ```
     [ -, -, -, 113, -, -, -, -, -, -, - ]
     ```

   - **Key 117**: $117 \mod 11 = 7$

     ```
     [ -, -, -, 113, -, -, -, 117, -, -, - ]
     ```

   - **Key 97**: $97 \mod 11 = 9$

     ```
     [ -, -, -, 113, -, -, -, 117, -, 97, - ]
     ```

   - **Key 100**: $100 \mod 11 = 1$

     ```
     [ -, 100, -, 113, -, -, -, 117, -, 97, - ]
     ```

   - **Key 114**: $114 \mod 11 = 4$

     ```
     [ -, 100, -, 113, 114, -, -, 117, -, 97, - ]
     ```

   - **Key 108**: $108 \mod 11 = 9$, collision at 9, quadratic probing to slot 10.

     ```
     [ -, 100, -, 113, 114, -, -, 117, -, 97, 108 ]
     ```

   - **Key 116**: $116 \mod 11 = 6$

     ```
     [ -, 100, -, 113, 114, -, 116, 117, -, 97, 108 ]
     ```

   - **Key 105**: $105 \mod 11 = 6$, collision at 6, quadratic probing to slot 0 (continuing the probe cycle).

     ```
     [ 105, 100, -, 113, 114, -, 116, 117, -, 97, 108 ]
     ```

- **Key 99**: $99 \mod 11 = 0$, collision at 0, quadratic probing to slot 5.

  `[ 105, 100, -, 113, 114, 99, 116, 117, -, 97, 108 ]`

3. **Final Hash Table Content**:

   `[ 105, 100, -, 113, 114, 99, 116, 117, -, 97, 108 ]`

Here, `-` indicates an empty slot.

(15) Considering a sequence of keys: `10, 4, 9, 8, 12, 15, 3, 5, 14, 18` for inserting into a heap. Please draw the result after insetring all these keys into an empty min heap. You need to list all the insertion steps and the final results.

Insert 10: Start with the heap [10].

Insert 4: Swap 4 and 10 since 4 < 10. Heap becomes [4, 10].

Insert 9: No swap needed, as 9 > 4. Heap is [4, 10, 9].

Insert 8: 8 swaps with 10 because 8 < 10. Heap changes to [4, 8, 9, 10].

Insert 12: 12 is greater than 8, no swap needed. Heap now [4, 8, 9, 10, 12].

Insert 15: 15 > 9, remains in place. Heap is [4, 8, 9, 10, 12, 15].

Insert 3: 3 < 9, swap 3 and 4. 3 < 4, swap 3 and 4. Heap becomes [3, 8, 4, 10, 12, 15, 9].

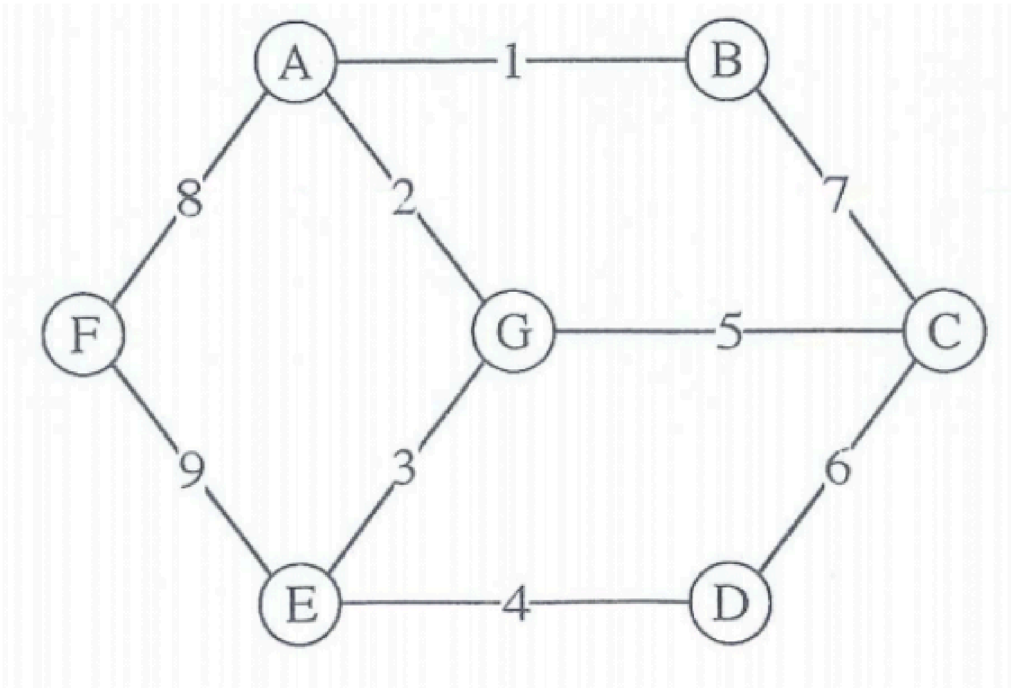
Insert 5: 5 < 10, swap 5 and 10. 5 < 8, swap 5 and 8. Heap updates to [3, 5, 4, 8, 12, 15, 9, 10].

Insert 14: 14 is larger than 8, stays in place. Heap is [3, 5, 4, 8, 12, 15, 9, 10, 14].

Insert 18: 18 > 12, no swap needed. Final heap [3, 5, 4, 8, 12, 15, 9, 10, 14, 18].

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

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



Show the order in which the edges are added to the minimum spanning tree using Prim's algorithm and the total cost of minimum spanning tree assume we start from vertext A.

1. **Starting at Vertex A**:
    - Initially add the smallest edge connected to A, which is (A-B) with weight 1.
2. **Vertices in MST**: {A, B}
    - Possible edges to add are (A-F) (8), (A-G) (2), and (B-C) (7).
    - Add (A-G) with weight 2.
3. **Vertices in MST**: {A, B, G}
    - Possible edges to add are (A-F) (8), (B-C) (7), (G-E) (3), (G-C) (5).
    - Add (G-E) with weight 3.
4. **Vertices in MST**: {A, B, G, E}
    - Possible edges to add are (A-F) (8), (B-C) (7), (G-C) (5), (E-D) (4), (E-F) (9).
    - Add (E-D) with weight 4.
5. **Vertices in MST**: {A, B, G, E, D}
    - Possible edges to add are (A-F) (8), (B-C) (7), (G-C) (5), (D-C) (6), (E-F) (9).
    - Add (G-C) with weight 5.
6. **Vertices in MST**: {A, B, G, E, D, C}
    - The remaining vertices not in the MST is (F). The edges to consider are (A-F) (8) and (E-F) (9).
    - Add (A-F) with weight 8.

The revised MST edges and their corresponding weights are:

- (A-B) with weight 1
- (A-G) with weight 2
- (G-E) with weight 3
- (E-D) with weight 4
- (G-C) with weight 5
- (A-F) with weight 8

Total cost of the MST is (1 + 2 + 3 + 4 + 5 + 8 = 23).

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

```
Preorder: ABEFCD
Inorder: CBDAEF
```

No.

The preorder traversal ABEFCD tells us that node A is the root. The inorder traversal CBDAEF tell us that nodes CBD (in the left of A) are in the left subtree and nodes EF (in the right of A) are in the right subtree. However, notice that the right subtree cannot be decomposed because nodes CBD are not contiguous in the preorder traversal. We cannot find the root of this subtree!