QUIZ!

Question 1

Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced)?

- Quick Sort
- Merge Sort
- Heap Sort
- Insertion Sort

Answer

Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced)?

- Quick Sort
- Merge Sort
- Heap Sort
- Insertion Sort

Question 2

Which of the following is not true about comparison based sorting algorithms?

- The minimum possible time complexity of a comparison based sorting algorithm is O(n log n) for a random input array
- Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared
- Counting Sort is not a comparison based sorting algorithm
- Heap Sort is not a comparison based sorting algorithm.

Answer

Which of the following is not true about comparison based sorting algorithms?

- The minimum possible time complexity of a comparison based sorting algorithm is O(n log n) for a random input array
- Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared
- Counting Sort is not a comparison based sorting algorithm
- **Heap Sort is not a comparison based sorting algorithm**

Question 3

Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?

- 1
- 2
- 3 or 4
- 5 or 6

Answer

Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?

- 1
- 2
- 3 or 4
- 5 or 6

Question 4

Which sorting algorithms is most efficient to sort string consisting of ASCII characters?

- Quick Sort
- Heap Sort
- Merge Sort
- Counting Sort

Answer

Which sorting algorithms is most efficient to sort string consisting of ASCII characters?

- Quick Sort
- Heap Sort
- Merge Sort
- Counting Sort

Question 5

What are the correct intermediate steps of the following data set when it is being sorted with the Insertion sort? 15,20,10,18

- ✗ 15,20,10,18 | 10,15,20,18 | 10,15,18,20 | 10,15,18,20
- ✗ 15,18,10,20 | 10,18,15,20 | 10,15,18,20 | 10,15,18,20
- ✗ 15,10,20,18 | 15,10,18,20 | 10,15,18,20
- ✗ 10, 20,15,18 | 10,15,20,18 | 10,15,18,20

Answer

What are the correct intermediate steps of the following data set when it is being sorted with the Insertion sort? 15,20,10,18

- × **15,20,10,18 | 10,15,20,18 | 10,15,18,20 | 10,15,18,20**
- × 15,18,10,20 | 10,18,15,20 | 10,15,18,20 | 10,15,18,20
- × 15,10,20,18 | 15,10,18,20 | 10,15,18,20
- × 10,20,15,18 | 10,15,20,18 | 10,15,18,20

Question 6

Consider the following lists of partially sorted numbers. The double bars represent the sort marker. How many comparisons and swaps are needed to sort the next number using Insertion Sort? [1 3 4 5 8 9 || 2]

- ✕ 5 Comparisons, 4 Swaps
- ✕ 4 Comparisons, 5 Swaps
- ✕ 6 Comparisons, 5 Swaps
- ✕ 5 Comparisons, 6 Swaps

Answer

Consider the following lists of partially sorted numbers. The double bars represent the sort marker. How many comparisons and swaps are needed to sort the next number using Insertion Sort? [1 3 4 5 8 9 || 2]

- ✕ 5 Comparisons, 4 Swaps
- ✕ 4 Comparisons, 5 Swaps
- ✕ **6 Comparisons, 5 Swaps**
- ✕ 5 Comparisons, 6 Swaps

Question 7

If all the elements in an input array is equal for example {1,1,1,1,1,1}, What would be the running time of the Insertion Algorithm?

- $O(2n)$
- $O(n^2)$
- $O(n)$
- None of the above

Answer

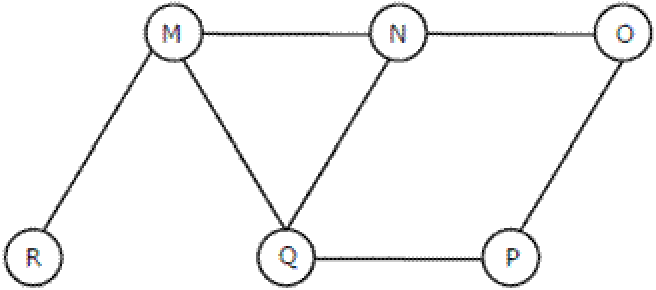
If all the elements in an input array is equal for example {1,1,1,1,1,1}, What would be the running time of the Insertion Algorithm?

- $O(2n)$
- $O(n^2)$
- **$O(n)$**
- None of the above

Question 8

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is

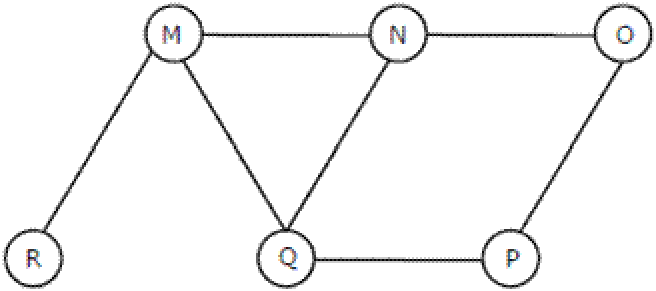
- MNOPQR
- NQMPOR
- QMNPRO
- QMNPOR



Answer

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is

- ✕ MNOPQR
- ✕ NQMPOR
- ✕ QMNPRO
- ✕ QMNPOR



Question 9

Let G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statements is always true? (GATE CS 2000)

- ✕ {u,v} must be an edge in G, and u is a descendant of v in T
- ✕ u,v} must be an edge in G, and v is a descendant of u in T
- ✕ If {u,v} is not an edge in G then u is a leaf in T
- ✕ If {u,v} is not an edge in G then u and v must have the same parent in T

Answer

Let G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statements is always true? (GATE CS 2000)

- ✕ {u,v} must be an edge in G, and u is a descendant of v in T
- ✕ u,v} must be an edge in G, and v is a descendant of u in T
- ✕ **If {u,v} is not an edge in G then u is a leaf in T**
- ✕ If {u,v} is not an edge in G then u and v must have the same parent in T

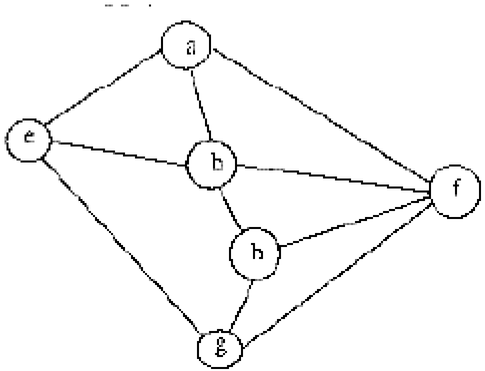
Question 10

Consider the following graph. Among the following sequences:

I) a b e g h f   II) a b f e h g   III) a b f h g e   IV) a f g h b e

Which are depth first traversals of the above graph? (GATE CS 2003)

- I, II and IV only
- I and IV only
- II, III and IV only
- I, III and IV only



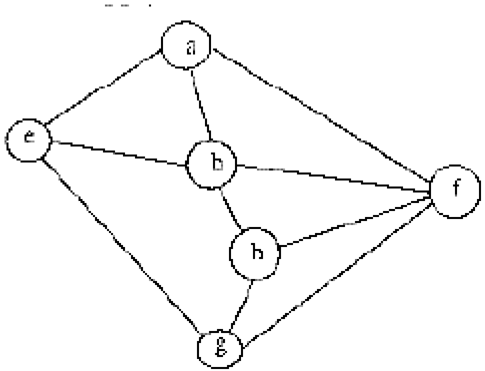
Answer

Consider the following graph. Among the following sequences:

I) a b e g h f   II) a b f e h g   III) a b f h g e   IV) a f g h b e

Which are depth first traversals of the above graph? (GATE CS 2003)

- I, II and IV only
- I and IV only
- II, III and IV only
- I, III and IV only



Question 11

Make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program. Which of the following standard graph algorithms is used by Make.

- Strongly Connected Components
- Topological Sorting
- Breadth First Search
- None of the above

Answer

Make is a utility that automatically builds executable programs and libraries from source code by reading files called makefiles which specify how to derive the target program. Which of the following standard graph algorithms is used by Make.

- Strongly Connected Components
- Topological Sorting
- Breadth First Search
- None of the above

Question 12

Given two vertices in a graph s and t, which of the two traversals (BFS and DFS) can be used to find if there is path from s to t?

- Only BFS
- Only DFS
- Both DFS and BFS
- Neither BFS nor DFS

Answer

Given two vertices in a graph s and t, which of the two traversals (BFS and DFS) can be used to find if there is path from s to t?

- ✕ Only BFS
- ✕ Only DFS
- ✕ Both DFS and BFS
- ✕ Neither BFS nor DFS

Question 13

Which of the following conditions is sufficient to detect cycle in a directed graph?

- There is an edge from currently being visited node to an already visited node
- There is an edge from currently being visited node to an ancestor of currently visited node in DFS forest
- Every node is seen twice in DFS
- None of the above

Answer

Which of the following conditions is sufficient to detect cycle in a directed graph?

- There is an edge from currently being visited node to an already visited node
- **There is an edge from currently being visited node to an ancestor of currently visited node in DFS forest**
- Every node is seen twice in DFS
- None of the above

Question 14

Is the following statement true/false?

If a DFS of a directed graph contains a back edge, any other DFS of the same graph will also contain at least one back edge.

- True
- False

Answer

Is the following statement true/false?

If a DFS of a directed graph contains a back edge, any other DFS of the same graph will also contain at least one back edge.

- True
- False

Question 15

Is the following statement true/false?

A DFS of a directed graph always produces the same number of tree edges, i.e., independent of the order in which vertices are considered for DFS.

- True
- False

Answer

Is the following statement true/false?

A DFS of a directed graph always produces the same number of tree edges, i.e., independent of the order in which vertices are considered for DFS.

- ✕ True
- ✕ False

Question 16

If the DFS finishing time f[u] > f[v] for two vertices u and v in a directed graph G, and u and v are in the same DFS tree in the DFS forest, then u is an ancestor of v in the depth first tree.

- True
- False

Answer

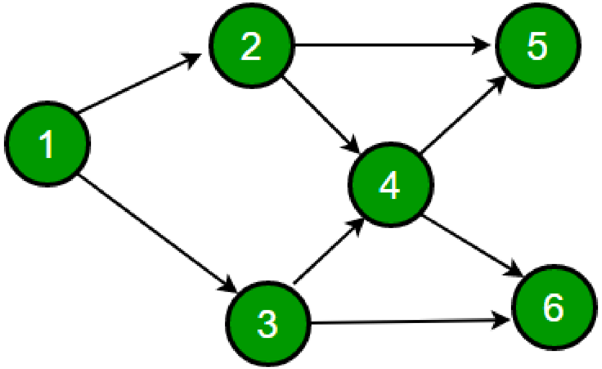
If the DFS finishing time f[u] > f[v] for two vertices u and v in a directed graph G, and u and v are in the same DFS tree in the DFS forest, then u is an ancestor of v in the depth first tree.

- True
- False

Question 17

Consider the DAG with Consider V = {1, 2, 3, 4, 5, 6}, shown below. Which of the following is NOT a Topological Ordering?

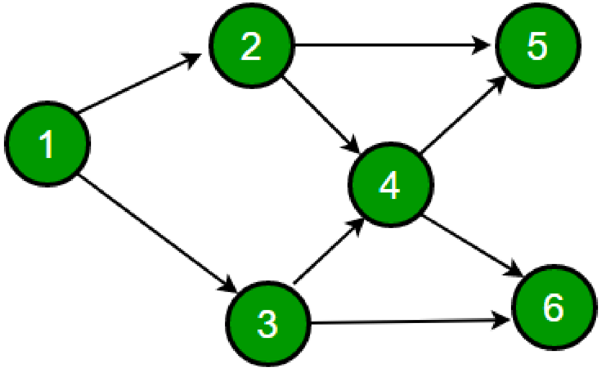
- 1 2 3 4 5 6
- 1 3 2 4 5 6
- 1 3 2 4 6 5
- 3 2 4 1 6 5



Answer

Consider the DAG with Consider V = {1, 2, 3, 4, 5, 6}, shown below. Which of the following is NOT a Topological Ordering?

- 1 2 3 4 5 6
- 1 3 2 4 5 6
- 1 3 2 4 6 5
- 3 2 4 1 6 5



Question 18

Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time on Depth First Search of G? Assume that the graph is represented using adjacency matrix.

- O(n)
- O(m+n)
- $O(n^2)$
- O(mn)

Answer

Let G be a graph with n vertices and m edges. What is the tightest upper bound on the running time on Depth First Search of G? Assume that the graph is represented using adjacency matrix.

- ✕ $O(n)$
- ✕ $O(m+n)$
- ✕ **$O(n^2)$**
- ✕ $O(mn)$

Question 19

Consider the tree edges of a BFS traversal from a source node W in an unweighted, connected, undirected graph. The tree T formed by the tree edges is a data structure for computing.

- ✕ the shortest path between every pair of vertices
- ✕ the shortest path from W to every vertex in the graph
- ✕ the shortest paths from W to only those nodes that are leaves of T
- ✕ the longest path in the graph

Answer

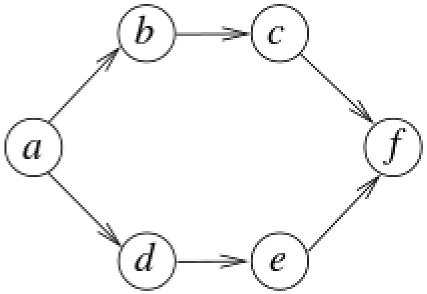
Consider the tree edges of a BFS traversal from a source node W in an unweighted, connected, undirected graph. The tree T formed by the tree edges is a data structure for computing.

- the shortest path between every pair of vertices
- **the shortest path from W to every vertex in the graph**
- the shortest paths from W to only those nodes that are leaves of T
- the longest path in the graph

Question 20

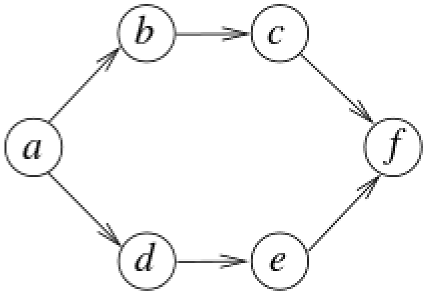
Consider the following directed graph. The number of different topological orderings of the vertices of the graph is: (GATE 2016)

- 1
- 2
- 4
- 6



Answer

Consider the following directed graph. The number of different topological orderings of the vertices of the graph is: (GATE 2016)



- 1
- 2
- 4
- 6