Stack and Queue MCQ

# Question 1

For the following algorithm, find a recurrence equation for the number of multiplications as a function of N, and represent the asymptotic growth rate of the solution to equation in θ notation. To simplify the analysis, you may assume that N is a power of two; that is N = 2k for some integer k. These functions compute X to the power N, where N is a positive integer.

*int A5Q (int X, int N)*

*{*

*int HALF;*

*if (N==1) return X;*

*else {*

*HALF=N/2;*

*if (2\*HALF==N) // if N is even*

*return A5Q (X, HALF) \* A5Q (X, HALF);*

*else return A5Q (X, HALF) \* A5Q (X, HALF) \* X;*

*}*

*}*

1. θ(log(n))
2. θ(n)
3. θ(n2)
4. θ(2n)

# Question 2

If f(n) = log10n, then which is/are correct statements(s)?

1. f(n)∈Ω(log2n)
2. ln n2 is the asymptotic lower bound of f(n)
3. f(n)∈Ο(n log n)
4. A
5. A and B
6. A and C
7. All of the above

# Question 3

For the following pairs of functions f(n) and g(n),

f(n) = n10

g(n) = 2n/2

,which of the following statement(s) is/are correct?

1. f(n) = Ο(g(n))
2. g(n) = Ο(f(n))
3. f(n) = Ω(g(n))
4. g(n) = Ω(f(n))
5. A
6. B, C
7. A, D
8. D
9. B
10. None of the above

# Question 4

If f(n) = e2n, then which is/are correct statement(s)?

1. f(n)∈Ο(32n)
2. f(n)∈Ω(2n)
3. f(n)∈θ(3n)
4. A
5. A and B
6. A and C
7. All of the above

Trees MCQ

Question 1

The average number of key comparisons done in a successful sequential search in a binary tree with n node is

1. log n
2. (n-1)/2
3. n/2
4. (n+1)/2
5. n\*(n+1)/2

Question 2

The average-case time complexity for binary search is

1. Ω(n log (n))
2. Ω(log (n))
3. Ο(log (n))
4. Ο(n log (n))
5. C is correct
6. B and C are correct
7. B, C and D are correct
8. A, B and C are correct

Question 3

Which of the following is not correct?

1. A linked list can be considered as a tree structure
2. A graph is a special case of a tree structure
3. Only the root of a binary tree has no parent
4. None of the above

Question 4

If the in-order traversal of a binary tree is E, C, D, B, F, A, J, H, G and the post-order traversal is E, D, C, F, B, J, H, G, A, what is the corresponding pre-order traversal of this binary tree?

1. A, B, C, E, F, D, H, G, J
2. A, B, C, E, F, D, G, H, J
3. A, B, C, D, E, F, G, H, J
4. None of the above

Question 5

If the depth of a node is the number of edges from the node to the root of its ternary tree, the maximum number of nodes at depth k is

1. 3k
2. 3(k-1)
3. 3(k+1)
4. 3k+1

Hash Table and Graph Representation MCQ

Question 1

How many nodes in the longest linked list which are created in the hash table when using closed addressing hashing with hash function:

h(key) = key mod 3, and input keys 3, 5, 6, 10, 12, 14, 16

1. 2
2. 3
3. 4
4. 5

Question 2

What are the first three probe slots that double hashing uses when

h(key) = key mod 11

d(key) = (key mod 7) – 7

and key 19?

1. Slot 8, Slot 6, Slot 4
2. Slot 8, Slot 10, Slot 12
3. Slot 6, Slot 4, Slot 2
4. Slot 10, Slot 12, Slot 14

Question 3

What are the first three probe sots that double hashing uses when

h(key) = key mod 11

d(key) = 7 – (key mod 7)

and key 19?

1. Slot 8, Slot 10, Slot 12
2. Slot 10, Slot 12, Slot 14
3. Slot 8, Slot 6, Slot 4
4. Slot 6, Slot 4, Slot 2

Question 4

Is the following graph strongly connected?

Diagram

Description automatically generated

1. Yes
2. No
3. Undecidable

Question 5

Which of the following statements is true?

1. The access time for the adjacency list is faster than the matrix
2. The space complexity for the adjacency list is higher than adjacency matrix
3. If the number of vertices is more than 2, a complete graph is also a cyclic graph
4. All connected graphs are also complete graphs

BFS and DFS MCQ

Question 1

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 after visiting u in the traversal. Which of the following statements is always true?

1. {u,v} must be an edge in G, and u is a descendant of v in T
2. If {u,v} is not an edge in G then u and v must have the same parent in T
3. If {u,v} is not an edge in G then u is a leaf in T
4. {u,v} must be an edge in G, and v is a descendant of u in T

Question 2

Traverse of a graph is different from a tree because

1. BFS of a graph uses queue, but a time efficient BFS of a tree is recursive.
2. There can be a loop in graph so we must maintain a visited flag for every vertex
3. DFS of a graph uses stack, but inorder traversal of a tree is recursive
4. All of the above
5. None of the above

Question 3

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

Diagram

Description automatically generated

1. MNOPQR
2. NQMPOR
3. QMNPRO
4. QMNPOR

Question 4

Consider the following graph

Chart

Description automatically generated

Among the following sequences:

1. a b e g h f
2. a b f e h g
3. a b f h g e
4. a f g h b e

Which are depth first traversals of the above graph?

1. I, II and IV only
2. I, III and IV only
3. I and IV only
4. II, III and IV only

Question 5

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 a path from s to t?

1. Only BFS
2. Only DFS
3. Both BFS and DFS
4. Neither BFS nor DFS

Backtracking and Dynamic Programming MCQ

Question 1

Given the following function:

F(n) = n \* F(n-1) + (n-1) \* F(n-2)

with F(0) = F(1) = 1.

Which technique can be used to find the value of F(n)?

1. Recursion
2. Dynamic Programming
3. A single “for” loop
4. All of the mentioned

Question 2

What happens when the backtracking algorithm reaches a complete solution?

1. It backtracks to the root
2. It continues searching for other possible solutions
3. It traverses from a different route
4. Recursively traverses through the same route

Question 3

Which algorithm cannot find all the solutions if there are many?

1. Brute force search
2. Backtracking
3. Dynamic Programming

Question 4

In what manner is a state-space tree for a backtracking algorithm constructed?

1. Depth-first search
2. Breadth-first search
3. Twice around the tree
4. Nearest neighbour first

Question 5

When a top-down approach of dynamic programming is applied to a problem, it usually \_\_\_.

1. Decreases both the time complexity and the space complexity
2. Decreases the time complexity and increases the space complexity
3. Increases the time complexity and decreases the space complexity
4. Increases both the time complexity and the space complexity