

# ADS TEST

TIME--45min ,40 question

[Sign in to Google](#) to save your progress. [Learn more](#)

\*Required

NAME \*

Khan Uzma Vikar Ahamad

EMAIL \*

uzmakhanvikar@gmail.com

CENTRE \*

KHARGHAR



Which of the following is the correct way to declare a multidimensional array 1 point  
in Java?

- ☐ a) `int[] arr;`
- ☐ b) `int arr[][];`
- ☒ c) `int[][]arr;`
- ☐ d) `int [[]] arr;`

Clear selection

In general, the index of the first element in an array is \_\_\_\_\_ 1 point

- ☒ a) 0
- ☐ b) -1
- ☐ c) 2
- ☐ d) 1

Clear selection



Elements in an array are accessed \_\_\_\_\_

1 point

- ☒ a) randomly
- ☐ b) sequentially
- ☐ c) exponentially
- ☐ d) logarithmically

Clear selection



1 point

```
public class array
{
    public static void main(String args[])
    {
        int []arr = {1,2,3,4,5};
        System.out.println(arr[5]);
    }
}
```

- ☐ a) 4
- ☐ b) 5
- ☒ c) ArrayIndexOutOfBoundsException
- ☐ d) InavlidInputException

Clear selection



What is the time complexity of the following code?

1 point

```
public boolean isBalanced(String exp)
{
    int len = exp.length();
    Stack<Integer> stk = new Stack<Integer>();
    for(int i = 0; i < len; i++)
    {
        char ch = exp.charAt(i);
        if (ch == '(')
            stk.push(i);
        else if (ch == ')')
        {
            if(stk.peek() == null)
            {
                return false;
            }
            stk.pop();
        }
    }
    return true;
}
```

- ☐ a)  $O(\log n)$
- ☒ b)  $O(n)$
- ☐ c)  $O(1)$
- ☐ d)  $O(n \log n)$

Clear selection



In a stack, if a user tries to remove an element from an empty stack it is called \_\_\_\_\_

1 point

- ☒ a) Underflow
- ☐ b) Empty collection
- ☐ c) Overflow
- ☐ d) Garbage Collection

Clear selection

Entries in a stack are “ordered”. What is the meaning of this statement?

1 point

- ☐ a) A collection of stacks is sortable
- ☐ b) Stack entries may be compared with the ‘<’ operation
- ☒ c) The entries are stored in a linked list
- ☐ d) There is a Sequential entry that is one by one

Clear selection



Which of the following is not the application of stack?

1 point

- ☐ a) A parentheses balancing program
- ☐ b) Tracking of local variables at run time
- ☐ c) Compiler Syntax Analyzer
- ☒ d) Data Transfer between two asynchronous process

Clear selection

What is the value of the postfix expression  $6\ 3\ 2\ 4\ +\ -\ *?$

1 point

- ☐ a) 1
- ☐ b) 40
- ☐ c) 74
- ☒ d) -18

Clear selection



The postfix form of the expression  $(A + B) * (C * D - E) * F / G$  is?

1 point

- ☐ a)  $AB + CD * E - FG / **$
- ☐ b)  $AB + CD * E - F **G /$
- ☒ c)  $AB + CD * E - *F *G /$
- ☐ d)  $AB + CDE * - *F *G /$

Clear selection

What data structure would you mostly likely see in non recursive implementation of a recursive algorithm?

1 point

- ☐ a) Linked List
- ☒ b) Stack
- ☐ c) Queue
- ☐ d) Tree

Clear selection





. The prefix form of  $A-B / (C * D ^ E)$  is?

1 point

- ☐ a)  $-/*^ACBDE$
- ☐ b)  $-ABCD*^ADE$
- ☒ c)  $-A/B*C^DE$
- ☐ d)  $-A/BC*^ADE$

Clear selection

The prefix form of an infix expression  $(p + q) - (r * t)$  is?

1 point

- ☐ a)  $+pq - *rt$
- ☐ b)  $- +pqr * t$
- ☒ c)  $- +pq * rt$
- ☐ d)  $- + * pqrt$

Clear selection



The data structure required for Breadth First Traversal on a graph is?

1 point

- ☐ a) Stack
- ☐ b) Array
- ☒ c) Queue
- ☐ d) Tree

Clear selection

Circular Queue is also known as \_\_\_\_\_

1 point

- ☒ a) Ring Buffer
- ☐ b) Square Buffer
- ☐ c) Rectangle Buffer
- ☐ d) Curve Buffer

Clear selection



If the elements "A", "B", "C" and "D" are placed in a queue and are deleted one at a time, in what order will they be removed? 1 point

- ☒ a) ABCD
- ☐ b) DCBA
- ☐ c) DCAB
- ☐ d) ABDC

Clear selection

A data structure in which elements can be inserted or deleted at/from both ends but not in the middle is? 1 point

- ☐ a) Queue
- ☐ b) Circular queue
- ☒ c) Dequeue
- ☐ d) Priority queue

Clear selection



A normal queue, if implemented using an array of size MAX\_SIZE, gets full when? 1 point

- ☒ a)  $\text{Rear} = \text{MAX\_SIZE} - 1$
- ☐ b)  $\text{Front} = (\text{rear} + 1) \bmod \text{MAX\_SIZE}$
- ☐ c)  $\text{Front} = \text{rear} + 1$
- ☐ d)  $\text{Rear} = \text{front}$

Clear selection

A linear collection of data elements where the linear node is given by means of pointer is called? 1 point

- ☒ a) Linked list
- ☐ b) Node list
- ☐ c) Primitive list
- ☐ d) Unordered list

Clear selection



In linked list each node contains a minimum of two fields. One field is data field to store the data second field is? 1 point

- ☐ a) Pointer to character
- ☐ b) Pointer to integer
- ☒ c) Pointer to node
- ☐ d) Node

Clear selection

What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list? 1 point

- ☐ a)  $O(1)$
- ☒ b)  $O(n)$
- ☐ c)  $\theta(n)$
- ☐ d)  $\theta(1)$

Clear selection



. What would be the asymptotic time complexity to find an element in the linked list?

1 point

- ☐ a)  $O(1)$
- ☒ b)  $O(n)$
- ☐ c)  $O(n^2)$
- ☐ d)  $O(n^4)$

Clear selection

Linked lists are not suitable for the implementation of \_\_\_\_\_

1 point

- ☐ a) Insertion sort
- ☐ b) Radix sort
- ☐ c) Polynomial manipulation
- ☒ d) Binary search

Clear selection



Which of the following code is used to create new node?

1 point

```
struct node
{
    int data;
    struct node * next;
}
typedef struct node NODE;
NODE *ptr;
```

- ☐ a) ptr = (NODE\*)malloc(sizeof(NODE));
- ☒ b) ptr = (NODE\*)malloc(NODE);
- ☐ c) ptr = (NODE\*)malloc(sizeof(NODE\*));
- ☐ d) ptr = (NODE)malloc(sizeof(NODE));

Clear selection



Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

1 point

- ☐ a) Insertion Sort
- ☐ b) Quick Sort
- ☒ c) Heap Sort
- ☐ d) Merge Sort

Clear selection

In the worst case, the number of comparisons needed to search a singly linked list of length  $n$  for a given element is?

1 point

- ☐ a)  $\log_2 n$
- ☐ b)  $n/2$
- ☐ c)  $\log_2 n - 1$
- ☒ d)  $n$

Clear selection





You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list? 1 point

- ☐ a) Delete the first element
- ☐ b) Insert a new element as a first element
- ☐ c) Delete the last element of the list
- ☒ d) Add a new element at the end of the list

Clear selection

How do you calculate the pointer difference in a memory efficient double linked list? 1 point

- ☐ a) head xor tail
- ☒ b) pointer to previous node xor pointer to next node
- ☐ c) pointer to previous node – pointer to next node
- ☐ d) pointer to next node – pointer to previous node

Clear selection



1 point

What is the functionality of the following piece of code?

```
public int function()  
{  
    Node temp = tail.getPrev();  
    tail.setPrev(temp.getPrev());  
    temp.getPrev().setNext(tail);  
    size--;  
    return temp.getItem();  
}
```

- ☒ a) Return the element at the tail of the list but do not remove it
- ☐ b) Return the element at the tail of the list and remove it from the list
- ☐ c) Return the last but one element from the list but do not remove it
- ☐ d) Return the last but one element at the tail of the list and remove it from the list

Clear selection



The optimal data structure used to solve Tower of Hanoi is \_\_\_\_\_

1 point

- ☐ a) Tree
- ☐ b) Heap
- ☐ c) Priority queue
- ☒ d) Stack

Clear selection

Which among the following is not a palindrome?

1 point

- ☐ a) Madam
- ☐ b) Dad
- ☐ c) Malayalam
- ☒ d) Maadam

Clear selection



How many children does a binary tree have?

0 points

- ☐ a) 2
- ☐ b) any number of children
- ☒ c) 0 or 1 or 2
- ☐ d) 0 or 1

Clear selection

What is/are the disadvantages of implementing tree using normal arrays?

1 point

- ☐ a) difficulty in knowing children nodes of a node
- ☐ b) difficult in finding the parent of a node
- ☒ c) have to know the maximum number of nodes possible before creation of trees
- ☐ d) difficult to implement

Clear selection



Can a tree stored in an array using either one of inorder or post order or pre order traversals be again reformed? 1 point

- ☒ a) Yes just traverse through the array and form the tree
- ☐ b) No we need one more traversal to form a tree
- ☐ c) No in case of sparse trees
- ☐ d) Yes by using both inorder and array elements

Clear selection

Disadvantages of linked list representation of binary trees over arrays? 1 point

- ☐ a) Randomly accessing is not possible
- ☐ b) Extra memory for a pointer is needed with every element in the list
- ☐ c) Difficulty in deletion
- ☒ d) Random access is not possible and extra memory with every element

Clear selection



Which of the following traversing algorithm is not used to traverse in a tree? 1 point

- ☐ a) Post order
- ☐ b) Pre order
- ☐ c) Post order
- ☒ d) Randomized

Clear selection

Level order traversal of a tree is formed with the help of 1 point

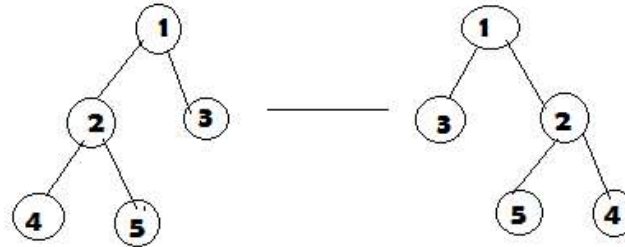
- ☒ a) breadth first search
- ☐ b) depth first search
- ☐ c) dijkstra's algorithm
- ☐ d) prims algorithm

Clear selection



1 point

9. What must be the missing logic below so as to print mirror of a tree as below as an example?



```
if(rootnode):
    mirror(rootnode-->left)
    mirror(rootnode-->right)

    //missing

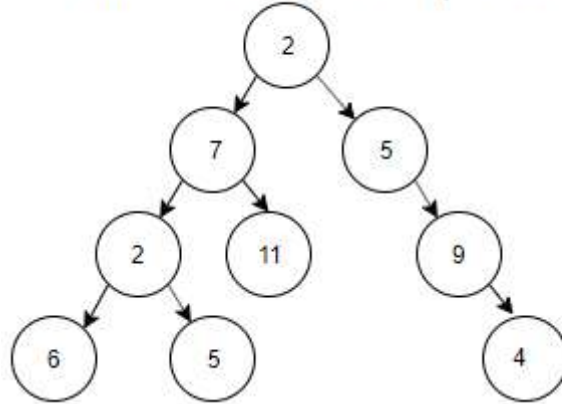
end
```

- ☐ a) swapping of left and right nodes is missing
- ☐ b) swapping of left with root nodes is missing
- ☐ c) swapping of right with root nodes is missing
- ☐ d) nothing is missing



1 point

1. For the tree below, write the pre-order traversal.



- ☐ a) 2, 7, 2, 6, 5, 11, 5, 9, 4
- ☐ b) 2, 7, 5, 2, 6, 9, 5, 11, 4
- ☐ c) 2, 5, 11, 6, 7, 4, 9, 5, 2
- ☐ d) 2, 7, 5, 6, 11, 2, 5, 4, 9





What is the time complexity of pre-order traversal in the iterative fashion? 1 point

- ☐ a)  $O(1)$
- ☐ b)  $O(n)$
- ☐ c)  $O(\log n)$
- ☐ d)  $O(n \log n)$

To obtain a prefix expression, which of the tree traversals is used? 1 point

- ☐ a) Level-order traversal
- ☐ b) Pre-order traversal
- ☐ c) Post-order traversal
- ☐ d) In-order traversal

Submit

Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms

