

Algorithm Analysis and Design Concepts

General

Data Structure and Algorithms

Analysis of Algorithms

Dashboard

Help Desk

FAQs

Performance Dashboard

Dashboard / Algorithm Analysis and Design Concepts / Analysis of Algorithms / Pre-Quiz

Started on	Tuesday, 18 February 2020, 1:20 AM
State	Finished
Completed on	Tuesday, 18 February 2020, 1:21 AM
Time taken	1 min 28 secs
Marks	4.50/5.00
Grade	90.00 out of 100.00
Feedback	Congratulations!! You have passed by securing more than 80%

Question 1

Correct
Mark 1.00
out of 1.00

Flag question

Jane has created a special type of linked list. That linked list contains no NULL values in its links. If so, what type of linked list is Jane has created?

Select one:

- ☐ None of these options
- ☐ Doubly Linked List
- ☒ Circular Linked List ✓
- ☐ Single Linked List

Your answer is correct.

The correct answer is: Circular Linked List

Question 2

Correct
Mark 1.00
out of 1.00

Flag question

Rearrange the below algorithm for computing n Factorial.
Input: n, an integer greater than or equal to 0
Output: n!

✓ procedure factorial(n)

✓ if n = 0 then

✓ return(1)

✓ return(n * factorial(n - 1))

✓ end factorial

Quiz navigation



Finish review

Your answer is correct.

Question 3

Correct

Mark 1.00
out of 1.00

Flag
question

What is the purpose of the following code snippets?

```
for (int i = 0; i < arr.length-1; i++)  
{  
    for (int j = i+1; j < arr.length; j++)  
    {  
        if( (arr[i].equals(arr[j])) && (i != j) )  
        {  
            System.out.println(arr[i]);  
        }  
    }  
}
```

Select one:

- ☐ Print the unique elements in the array
- ☐ Print the element with maximum frequency
- ☐ None of the these
- ☒ Print the duplicate elements in the array ✓

Your answer is correct.

The correct answer is: Print the duplicate elements in the array

Question 4

Correct

Mark 1.00
out of 1.00

Flag
question

Select the code snippet which performs unordered linear search iteratively?

Select one:

- ☐

```
public int UnorderedLinearSearch(int[] arr, int size, int data)  
{  
    int index=0;  
    for(int i = 0; i <= size; i++)  
    {  
        if(arr[i] == data)  
        {  
            index = i;  
            break;  
        }  
    }  
    return index;  
}
```
- ☐

```
public int UnorderedLinearSearch(int[] arr, int size, int data)
```

```

    {
        int index;
        for(int i = 0; i < size; i++)
        {
            if(arr[i] == data)
            {
                break;
            }
        }
        return index;
    }
}

```

- ☐ None of these options
- ☒ public int UnorderedLinearSearch(int[] arr, int size, int data)
- ```

{
 int index=0;
 for(int i = 0; i < size; i++)
 {
 if(arr[i] == data)
 {
 index = i;
 break;
 }
 }
 return index;
}

```

Your answer is correct.

The correct answer is: public int UnorderedLinearSearch(int[] arr, int size, int data)

```

{
 int index=0;
 for(int i = 0; i < size; i++)
 {
 if(arr[i] == data)
 {
 index = i;
 break;
 }
 }
 return index;
}

```

## Question 5

Partially correct

Mark: 0.50

Choose the scenario(s) when we need to use Linear search?

Select one or more:

Mark 0.50  
out of 1.00

Flag  
question

- ☐ Can use all the time
- ☐ None of these options
- ☐ When the list has only a few elements
- ☒ When performing a single search in an un-ordered list ✓

Your answer is partially correct.

You have correctly selected 1.

The correct answers are: When the list has only a few elements, When performing a single search in an un-ordered list

[Finish review](#)



Powered by Tekstac

