Data Structure and Algorithms Time Allowed 1 Hr (Intake 42)

IT is NOT ALLOWED to Use any Digital tools
IT is NOT ALLOWED to open any software
IT is NOT ALLOWED to Open any Browser TAB except that of the Exam TAB
ONLY ALLOWED to use HARD COPY

* Required

1

Enter Your Name (IN English) *

Youssef Ibrahim Salama

2

Select Your Track *

- Professional Web Development and BI
- [✓ Open Source Application Development]

3

What is the ' next ' field of structure node in the Queue?
(2 Points)

- a. Results into the storage of queue elements.
- [✓ b. Results into the storage of address of next node by holding the next element of queue.]
- c. Results into the memory allocation of data elements to next node.
- d. Results into the address allocation data elements to next node.

4

Linked lists are best suited ....
(2 Points)

- a. For relatively permanent collections of data
- [✓ b. For the size of the structure and the data in the structure are constantly changing]
- c. Data structure
- d. For none of the above situation

5

The operation of processing each element in the list is known as
(2 Points)

- a. Sorting
- b. Merging
- c. Inserting
- [✓ d. Traversal]

6

Each node in a linked list must contain at least
(2 Points)

- a. Three fields
- [✓ b. Two fields]
- c. Four fields
- d. Five fields

7

A Linear list in which the pointer points only to successive node is
(2 Points)

- [✓ a. Single linked list]
- b. Circular linked list
- c. Doubly linkedlist
- d. None of the above

8

The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called ......
(2 Points)

- a. Inserting
- b. Mixing
- [✓ c. Merging]
- d. Sharing

9

The worst case occurs in linear search algorithm when ......
(2 Points)

- a. Item is somewhere in the middle of the array
- b. Item is not in the array at all
- c. Item is the last element in the array
- [✓ d. Item is the last element in the array or item is not there at all]

10

Inorder traversing a tree resulted E A C K F H D B G, the PostOrder traversing of the same tree resulted E C K A H B G D F, So the PreOrder traversing would return:
(2 Points)

- a. FAEKCDBHG
- [✓ b. FAEKCDHGB]
- c. EAFKHDCBG
- d. FEAKDCHBG

11

Traversing a Binary Search tree Pre-Order resulted: F C B A D E H G I J, so the traversing Post order will result:
(2 Points)

- a. ABEDCGIJHF
- b. ABDECFHGIJ
- c. ABEDCGJIHF
- [✓ d. None of the above]

12

What does the following function do for a given Linked List with first node as head?

```
void fun1(struct node* head)
{
    if(head == NULL)
       return;
    fun1(head->next);
    printf("%d  ", head->data);
}
```

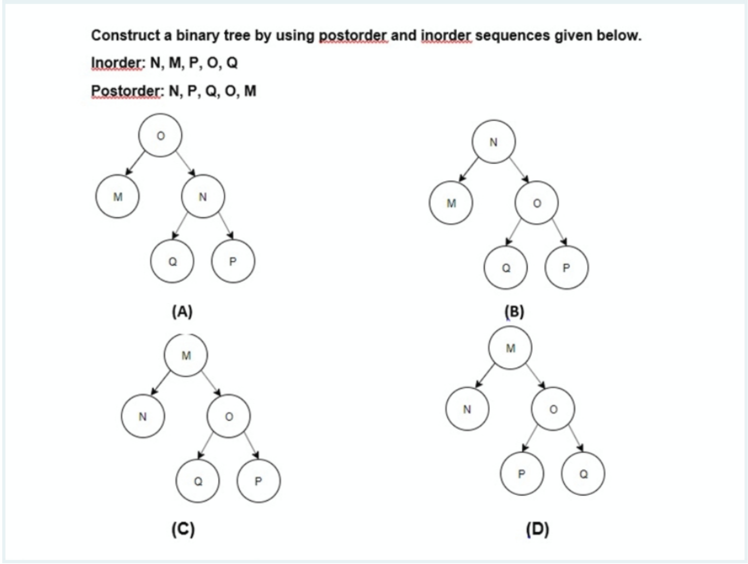
(2 Points)

- a. Prints all nodes of linked lists
- [✓ b. Prints all nodes of linked list in reverse order]
- c. Prints alternate nodes of Linked List
- d. Prints alternate nodes in reverse order

13

Question

(2 Points)



Construct a binary tree by using postorder and inorder sequences given below.

Inorder: N, M, P, O, Q

Postorder: N, P, Q, O, M

- A
- B
- C
- [✓ D]

14

What will be the output of the following program?

```
main()
{
    char str[]="san foundry";
    int len = strlen(str);
    int i;
    for(i=0;i
```

(2 Points)

- a. sanfoundry
- b. san foundry
- c. yrdnuof nas
- d. foundry nas

15

Which of the following code snippet performs linear search recursively?

```
A) LinearSearch(int[] a, n, key)
   {
       for(i=0;i
```

(2 Points)

- A
- B
- C
- D

16

Choose the appropriate code that does binary search using recursion.

```
A) public int recursive(int arr[], int low, int high, int key)
   {
       int mid = low + (high - low)/2;
       if(arr[mid] == key)
       {
              return mid;
       }
       else if(arr[mid] < key)
       {
              return recursive(arr,mid+1,high,key);
       }
       else
       {
              return recursive(arr,low,mid-1,key);
       }
   }
B) public int recursive(int arr[], int low, int high, int key)
   {
       int mid = low + (high + low)/2;
       if(arr[mid] == key)
       {
              return mid;
       }
       else if(arr[mid] < key)
       {
              return recursive(arr,mid-1,high,key);
       }
       else
       {
              return recursive(arr,low,mid+1,key);
       }
   }
C) public int recursive(int arr[], int low, int high, int key)
   {
       int mid = low + (high - low)/2;
       if(arr[mid] == key)
       {
              return mid;
       }
       else if(arr[mid] < key)
       {
              return recursive(arr,mid,high,key);
       }
       else
       {
              return recursive(arr,low,mid-1,key);
       }
   }
D) public int recursive(int arr[], int low, int high, int key)
   {
       int mid = low + ((high - low)/2)+1;
       if(arr[mid] == key)
       {
              return mid;
       }
       else if(arr[mid] < key)
       {
              return recursive(arr,mid,high,key);
       }
       else
       {
              return recursive(arr,low,mid-1,key);
       }
  }
```

(2 Points)

- [✓ A]
- B
- C
- D

17

Choose the correct code for merge sort.

```
A) void merge_sort(int arr[], int left, int right)
  {
     if (left > right)
     {
       int mid = (right-left)/2;
       merge_sort(arr, left, mid);
       merge_sort(arr, mid+1, right);
       merge(arr, left, mid, right); //function to merge sorted arrays
     }
  }
B) void merge_sort(int arr[], int left, int right)
   {
      if (left < right)
      {
         int mid = left+(right-left)/2;
         merge_sort(arr, left, mid);
         merge_sort(arr, mid+1, right);
         merge(arr, left, mid, right); //function to merge sorted arrays
      }
   }
C) void merge_sort(int arr[], int left, int right)
   {
      if (left < right)
      {
          int mid = left+(right-left)/2;
          merge(arr, left, mid, right); //function to merge sorted arrays
          merge_sort(arr, left, mid);
          merge_sort(arr, mid+1, right);

     }
   }
D) void merge_sort(int arr[], int left, int right)
   {
      if (left < right)
      {
          int mid = (right-left)/2;
          merge(arr, left, mid, right); //function to merge sorted arrays
          merge_sort(arr, left, mid);
          merge_sort(arr, mid+1, right);
      }
   }
```

(2 Points)

- ○ A
- [✓ B]
- ○ C
- ○ D

18

What is the output of following function for start pointing to first node of following linked list?

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

```
void fun(struct node* start)
{
   if(start == NULL)
       return;
   printf("%d  ", start->data);
   if(start->next != NULL )
       fun(start->next->next);
   printf("%d  ", start->data);
}
```

(2 Points)

- ○ a) 1 4 6 6 4 1
- ○ b) 1 3 5 1 3 5
- ○ c) 1 2 3 5
- [✓ d) 1 3 5 5 3 1]

19

The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node
{
    int value;
    struct node *next;
};
void rearrange(struct node *list)
{
    struct node *p, * q;
    int temp;
    if ((!list) || !list->next)
      return;
    p = list;
    q = list->next;
    while(q)
    {
         temp = p->value;
         p->value = q->value;
         q->value = temp;
         p = q->next;
         q = p?p->next:0;
    }
}
```
(2 Points)

- ○ a) 1, 2, 3, 4, 5, 6, 7
- [✓ b) 2, 1, 4, 3, 6, 5, 7]
- ○ c) 1, 3, 2, 5, 4, 7, 6
- ○ d) 2, 3, 4, 5, 6, 7, 1

20

Array implementation of Stack is not dynamic, which of the following statements supports this argument?
(2 Points)

- [✓ a) space allocation for array is fixed and cannot be changed during run-time]
- ○ b) user unable to give the input for stack operations
- ○ c) a runtime exception halts execution
- ○ d) improper program compilation

21

In linked list implementation of a queue, where does a new element be inserted?
(2 Points)

- ○ a) At the head of link list
- ○ b) At the centre position in the link list
- [✓ c) At the tail of the link list]
- ○ d) At any position in the linked list

22

Which of the following is not a disadvantage to the usage of array?
(2 Points)

- ○ a) Fixed size
- ○ b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
- ○ c) Insertion based on position
- [✓ d) Accessing elements at specified positions]

Submit

Never give out your password. Report abuse