Data Structures

1. struct node{
    int data;
    node *next;
}

The node definition above could be used for the implementation of a

(a) doubly linked list
(b) binary tree
(c) linked list
(d) heap
(e) all of the above

2. The following sequence of operations is performed on an initially empty stack (represented by the variable s):
s.push(4); s.push(7); s.push(3); s.push(8); s.pop(x); s.pop(x); s.push(2); s.push(5); s.pop(x);
Which of the following correctly represents the resultant state of the stack, top to bottom, and the final final of x?

   top -> bottom

(a) 2, 7, 4    x=5
(b) 5, 2, 8    x=3
(c) 2, 7, 4    x=2
(d) 8, 3, 7, 4 x=5
(e) 7, 4, 5    x=2

3. Insert the following integers into a binary search tree: 5, 12, 3, 1, 2
Which diagram correctly represents the resultant binary search tree?

4. If big-O notation is used to describe the efficiency of an algorithm which of the following would be least efficient?

(a) \(O(n^2)\)
(b) \(O(n \log n)\)
(c) \(O(n^{10})\)
(d) \(O(2^{1.8})\)
(e) \(O(n)\)
5. The average performance of Quicksort is:
   (a) $\Theta(n^2)$
   (b) $\Theta(n\log n)$
   (c) $\Theta(2^n)$
   (d) $\Theta(n)$
   (e) $\Theta(\log n)$

6. The value contained in any node of a heap is:
   (a) smaller than the value of either of its children (if any exist)
   (b) at least as large as the value of its left child but no larger than the value of its right child (if any exist)
   (c) at least as large as the value of either of its children (if any exist)
   (d) arbitrary
   (e) equal to $\log_2 n$ if there are $n$ nodes in the heap

![Binary Tree Diagram](image)

7. Consider the binary tree shown above. In what order will the vertices be visited if a preorder traversal is used?
   (a) 1, 2, 5, 6, 3, 4
   (b) 1, 5, 3, 6, 2, 4
   (c) 1, 2, 3, 5, 6, 4
   (d) 1, 2, 5, 3, 6, 4
   (e) 5, 6, 3, 4, 2, 1

8. Which of the following correctly defines the function insert for the class queue? (A queue represented by a linked list of nodes, of the form, pointed to by the variable front, and the value removed from the queue is returned in an outout parameter of type integer.)
   (a) void queue::insert(int item)
       {item = front->data; node *p = front; front = front->next; delete p;
        if(front == nil) rear=nil;}
   (b) void queue::insert(int item)
       {node *p = front; front = front->next; item = front->data; delete p;
        if(front == nil) rear=nil;}
   (c) void queue::insert(int item)
       {item = front->data; front = front->next; node *p = front; delete p;
        if(front == nil) rear=nil;}
   (d) void queue::insert(int item)
       {node *p = front; delete p; item = front->data; front = front->next;
        if(front == nil) rear=nil;}

2
9. int func(int n) {
    if (n<=1) return 1;
    else return n*func(n-1);
}

The recursive function given above computes
(a) the nth Fibonacci number
(b) n factorial
(c) $2^n$
(d) $\log_2 n$
(e) the sum of the first n positive integers

10. struct node{
    float data;
    node *next;
}

float sum_data(node *head){
    float sum = 0;
    node *temp = head;
    while (temp != NULL){
        sum = sum + temp->data;
        temp = __________;
    }
    return sum;
}

The above function computes the sum of all of the data fields in a linked list with the given structure. The blank should contain:
(a) temp->data
(b) head->data
(c) head->next
(d) temp->next
(e) next
11. The diagram above is a graphical representation of:
   (a) a singly linked list
   (b) a doubly linked list
   (c) a circularly linked list
   (d) a binary search tree
   (e) a heap

12. The average performance of Binary Search of a list with \( n \) items is:
   (a) \( \Theta(n^2) \)
   (b) \( \Theta(n \log n) \)
   (c) \( \Theta(2^n) \)
   (d) \( \Theta(n) \)
   (e) \( \Theta(n \log n) \)