CSE 202 Midterm 1 Part (b)

Olcay Taner YILDIZ

I. Multiple Choice Questions (50 points)

1. Which of the following functions has the largest growth?

a) n^2
b) n
c) $n\log(n)$
d) 2^n
e) n^6

2. What is the time complexity of the following fragment of code?

```
for (int i =1; i <= N; i++)
  for (int j =1; j <= i; j++)
     sum += i * i;
```

a) $\mathcal{O}(N)$
b) $\mathcal{O}(1)$
c) $\mathcal{O}(N^2)$
d) $\mathcal{O}(\log N)$
e) $\mathcal{O}(\sqrt{N})$

3. What is the time complexity of the following fragment of code?

```
int i = 1;
while (i < N){
   i = i * 2;
}
```

a) $\mathcal{O}(N)$
b) $\mathcal{O}(1)$
c) $\mathcal{O}(N^2)$
d) $\mathcal{O}(\sqrt{N})$
e) $\mathcal{O}(\log N)$

4. What is the value of N after the following code frament is executed?

```
int sum = 0;
for (i = 0; i < N; i++)
   for (j = 0; j < N * N; j++)
       sum++;
```

a) $N^2 + N^3$
b) N^3
c) N^2
d) N
e) N^4

5. Given the time complexity equation for a recursive algorithm as $T(N) = 4T(N/2) + \mathcal{O}(N^2)$. What is the time complexity of that algorithm?

a) $\mathcal{O}(N^2 \log N)$
b) $\mathcal{O}(N^2)$
c) $\mathcal{O}(N^3)$
d) $\mathcal{O}(N \log N)$
e) $\mathcal{O}(N)$

6. Given the time complexity equation for a recursive algorithm as $T(N) = T(N/2) + \mathcal{O}(N)$. What is the time complexity of that algorithm?

a) $\mathcal{O}(N^2 \log N)$
b) $\mathcal{O}(N^2)$
c) $\mathcal{O}(N^3)$
d) $\mathcal{O}(N \log N)$
e) $\mathcal{O}(N)$

7. If you do not use a last pointer in the singly linked list implementation, what will be the time complexity of insertFirst?

a) $\mathcal{O}(1)$
b) $\mathcal{O}(N)$
c) $\mathcal{O}(\log N)$
d) $\mathcal{O}(N^2)$
e) $\mathcal{O}(N \log N)$

8. If you do not use a last pointer in the singly linked list implementation, what will be the time complexity of insertLast?

a) $\mathcal{O}(1)$
b) $\mathcal{O}(N)$
c) $\mathcal{O}(\log N)$
d) $\mathcal{O}(N^2)$
e) $\mathcal{O}(N \log N)$

9. Given the following data a = (4-3-8-9-2) as a singly linked list. What will be the linked list after the following operations?

```
a.insertFirst(4)
a.insertFirst(9)
a.deleteLast
a.insertLast(3)
a.deleteFirst()
```

a) 4-3-8-9-2
b) 4-4-8-9-2-3
c) 4-3-8-9-3
d) 4-4-3-8-9-3
e) 4-3-8-9-2-3

10. What is the time complexity of the deleteLast for singly linked list, doubly linked list, and circular linked list?

a) $\mathcal{O}(1)$, $\mathcal{O}(1)$, $\mathcal{O}(1)$
b) $\mathcal{O}(N)$, $\mathcal{O}(N)$, $\mathcal{O}(N)$
c) $\mathcal{O}(N)$, $\mathcal{O}(1)$, $\mathcal{O}(1)$
d) $\mathcal{O}(N)$, $\mathcal{O}(1)$, $\mathcal{O}(N)$
e) $\mathcal{O}(1)$, $\mathcal{O}(1)$, $\mathcal{O}(N)$

11. Given that there are more than one node in a single linked list, which of the following code fragment **deletes** the **first** node?

a)
```
first = first.next;
```
b)
```
first.next = first;
```
c)
```
first = first.next;
last = first;
```
d)
```
last = last.next;
first = last;
```
e)
```
last = last.next;
```

12. Given that there are more than one node in a single linked list, which of the following code fragment **inserts** a new node at the **end** of the list?

a)
```
last.next = newNode;
```
b)
```
last = newNode;
last.next = newNode;
```
c)
```
last.next = newNode;
last = newNode;
```
d)
```
last = newNode;
```
e)
```
last = last.next;
last = newNode;
```

13. Given that there are more than one node in a doubly linked list, which of the following code fragment **deletes** the **last** node?

a)
```
last = last.previous;
```
b)
```
last = last.previous;
last.next = null;
```
c)
```
last.next = null;
last = last.previous;
```
d)
```
last.next = last;
```
e)
```
last = last.next;
last.previous = last;
```

14. Which of the following code fragment **inserts** a new node into the **middle** of a doubly linked list, where **previous** shows the previous node after new node is inserted?

a)
```
newNode.next = previous.next;
newNode.previous = previous;
previous.next = newNode;
previous.next.previous = newNode;
```
b)
```
previous.next.previous = newNode;
previous.next = newNode;
```
c)
```
newNode.previous = previous;
newNode.next = previous.next;
```
d)
```
previous.next.previous = newNode;
previous.next = newNode;
newNode.next = previous.next;
newNode.previous = previous;
```
e)
```
newNode.next = previous.next;
newNode.previous = previous;
previous.next.previous = newNode;
previous.next = newNode;
```

15. Which of the following orderings can not be obtained for 1,2,3,4 if those numbers must be pushed in that order but can be popped in any order?

a) 1-2-3-4
b) 2-1-4-3
c) 1-3-2-4
d) 1-3-4-2
e) 3-2-1-4

16. What are the time complexities of push and pop operations for array and linked list implementations?

a) Array: $\mathcal{O}(N)$, $\mathcal{O}(1)$, List: $\mathcal{O}(1)$, $\mathcal{O}(1)$
b) Array: $\mathcal{O}(N)$, $\mathcal{O}(1)$, List: $\mathcal{O}(N)$, $\mathcal{O}(1)$
c) Array: $\mathcal{O}(N)$, $\mathcal{O}(N)$, List: $\mathcal{O}(1)$, $\mathcal{O}(1)$
d) Array: $\mathcal{O}(1)$, $\mathcal{O}(1)$, List: $\mathcal{O}(1)$, $\mathcal{O}(1)$
e) Array: $\mathcal{O}(1)$, $\mathcal{O}(N)$, List: $\mathcal{O}(1)$, $\mathcal{O}(N)$

17. Given the following data a = (4-3-8-9-2) as a stack, where 4 is the at the top of the stack. What will be the stack after the following operations?

```
a.push(3)
a.pop
a.pop
a.pop
a.push(4)
```

a) 4-8-9-2
b) 8-3-9-2
c) 8-4-9-2
d) 3-8-9-2
e) 4-4-9-2

18. Which of the following code fragment insert a new element onto the stack for the linked list implementation?

a)
```
top = newNode;
newNode.next = top;
```
b)
```
newNode.next = top;
top = newNode;
```
c)
```
top = newNode;
```
d)
```
newNode.next = top;
```
e)
```
newNode = top;
```

19. Given that the stack is not full, which of the following code fragment insert a new element onto the stack for the array implementation?

a)
```
array[top] = element;
top++;
```
b)
```
top++;
```
c)
```
array[top] = element;
```
d)
```
array[top++] = element;
```
e)
```
top++;
array[top] = element;
```

20. Which of the following operations on which implementations can give stack overflow?

a) pop, array
b) pop, linked list
c) push, array
d) push, array
   pop, array
e) push, linked list