THE UNIVERSITY OF WESTERN AUSTRALIA

MID SEMESTER EXAMINATION
April 2018

DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

DATA STRUCTURES AND ALGORITHMS CITS2200

This Paper Contains:
- 6 Pages
- 10 Questions

Time allowed: Forty five minutes

Marks for this paper total 10.
Students should answer ALL Questions.
Q1. Which one of the following statements about the worst-case complexity of Insertion Sort is wrong?

(A) The worst-case complexity of Insertion Sort is $O(n^4)$.
(B) The worst-case complexity of Insertion Sort is $O(n^3)$.
(C) The worst-case complexity of Insertion Sort is $O(n \log n)$.
(D) The worst-case complexity of Insertion Sort is $O(n^2)$.

Q2. Which one of the following statements about the worst-case complexity of Quick Sort is correct?

(A) The worst-case complexity of Quick Sort is $O(\log n)$.
(B) The worst-case complexity of Quick Sort is $O(n^2)$.
(C) The worst-case complexity of Quick Sort is $O(n \log n)$
(D) The worst-case complexity of Quick Sort is $O(n)$.

Q3. Suppose $f(n)$ is $O(g(n))$, $g(n)$ is $O(h(n))$, and $h(n)$ is $O(f(n))$. Which of the following are possible functions for $f$, $g$ and $h$?

(A) $f(n) = \log^2 n$, $g(n) = n \log n$, $h(n) = n^2$.
(B) $f(n) = n^3$, $g(n) = n^2$, $h(n) = n \log n$.
(C) $f(n) = 5 \log n$, $g(n) = 1000 \log n$, $h(n) = \log n$.
(D) $f(n) = n^2$, $g(n) = n^4$, $h(n) = 2^n$. 
Q4. A (singly) linked implementation of a Queue contains the following instance variables:

- **front**: A reference to the front of the queue, that is, the end with the item that has been in the queue for the longest amount of time;
- **back**: A reference to the back of the queue, that is, the end with the item that has been added most recently.

The `enqueue` method can be implemented as:

```java
public void enqueue (Object a) {
    if (isEmpty()) {
        front=new Link(a,null);
        back=front;
    }

    <<missing code>>
}
```

**Note**: All operations in the queue must be able to operate in constant time.

Which of the following is a correct implementation of the missing code?

(A) else back.successor= new Link(a,null);
(B) else front= new Link(a,front);
(C) else {
    front.successor=new Link(a,front);
    front=front.successor;
}
(D) else {
    back.successor= new Link(a,null);
    back=back.successor;
}
Q5. Suppose \( f(n) = 2^{\log n}, g(n) = n^2, h(n) = n\sqrt{n}, k(n) = n\log n, p(n) = 2^n \). Which of the following is a correct ordering of these complexities in ascending order (smallest to largest)?

(A) \( f(n), h(n), g(n), p(n), k(n) \).
(B) \( f(n), k(n), h(n), g(n), p(n) \).
(C) \( h(n), k(n), f(n), g(n), p(n) \).
(D) \( p(n), k(n), f(n), g(n), h(n) \).

Q6. Consider the following figure:

Using the definition of the `Link` class from the lectures and labs, which one of the following codes transforms the first figure to the second figure?

(A) `first.successor.successor=first.successor`
(B) `first.successor.successor.successor=first.successor`
(C) `first.successor=first.successor.successor.successor`
(D) `first.successor.successor.successor=first.successor`
Q7. Consider the following figure:

Using the definition of the Link class from the lectures and labs, which one of the following codes transforms the first figure to the second figure?

(A)  
```java
char temp = first.successor.item;
first.successor.successor.item = temp;
first.successor.item = first.successor.successor.item;
first.successor.successor.successor = first;
```

(B)  
```java
char temp = first.successor.item;
first.successor.item = first.successor.successor.item;
first.successor.successor.item = temp;
first.successor.successor.successor = first;
```

(C)  
```java
char temp = first.item;
first.successor.siccessor.item = temp;
first.successor.item = first.successor.successor.item;
first.successor.successor.successor = first;
```

(D)  
```java
char temp = first.successor.item;
first.successor.item = first.successor.successor.item;
first.successor.successor.item = temp;
first.successor.successor.successor = first;
```
Q8. Which of the following statements is true?

(A) The worst case complexity of quicksort is $O(n \log n)$ and the average case complexity is $O(n^2)$.

(B) Both the worst case and the average case complexities of quicksort are $O(n^2)$.

(C) The average case complexity of quicksort is $O(n \log n)$ and the worst case complexity is $O(n \log n)$.

(D) The average case complexity of quicksort is $O(n \log n)$ and the worst case complexity is $O(n^2)$.

Q9. The correct recurrence equation for analysing the complexity of the Merge Sort algorithm is ($c$ is a constant):

(A) $T(n) = T(n^2) + cn$.

(B) $T(n) = T(n - 1) + cn$.

(C) $T(n) = 4T(n^2) + cn$.

(D) $T(n) = 2T(n^2) + cn$.

Q10. If there are $n$ objects in a list, the time complexity to delete the middle object is:

(A) $O(n \log n)$

(B) $O(n)$

(C) $O(n^2)$

(D) none of the above.

END OF PAPER