THE UNIVERSITY OF WESTERN AUSTRALIA

FIRST SEMESTER EXAMINATION
JUNE 2016

SCHOOL OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

DATA STRUCTURES AND ALGORITHMS CITS2200

This Paper Contains:
6 Pages
5 Questions

Time allowed : **TWO HOURS**
Reading time : **10 Minutes**

All questions carry equal marks.
Marks for this paper total 50.
Candidates should answer **ALL** Questions.

This page has been left blank intentionally.

Q1.

(a) Write pseudo code for the `merge sort` algorithm, explaining clearly the different parts of your code.

(5)

(b)

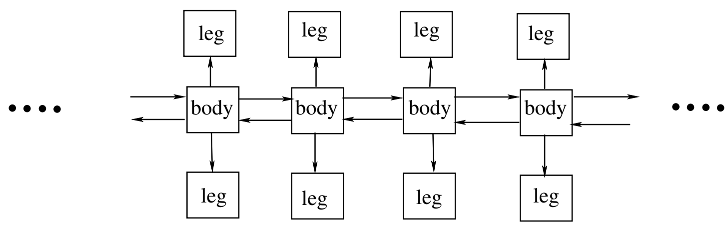
- What is the asymptotic time complexity of the `merge sort` algorithm?
- Write a recurrence relation for expressing the time complexity of the `merge sort` algorithm. You need not solve it.
- Simulate the algorithm you have written in part **(a)** on the following array for sorting it in ascending order:

  `9, 8, 7, 6, 5, 4, 3, 2`

(5)

Q2.

The `caterpillar` data structure looks like the following picture:



- Each `body` node is connected to two `leg` nodes, as shown in the picture. Each body node is connected to its predecessor and successor body nodes through double links.
- Each body node and each leg node can store an integer item.
- You can assume a `WindowLinked` class and an object `w` of this class. `w.link` indicates the position of a window on a body node (in other words, `w.link` is a reference to a body node).

(a)

- Write two class definitions `ListBody` and `ListLeg` for implementing the caterpillar data structure.
- Write a method `ConnectLegToBody(ListBody b, ListLeg l)` for connecting an object of class `ListBody` to an object of class `ListLeg`.

(5)

(b)

- Write a method `DeleteBefore(WindowLinked w)` to delete the node that is to the left of the node that the window is on (in other words, the `ListBody` node and its two legs to the left of the current window position are to be deleted).
- Write a method `Search(int i)` that searches for an integer `i` in a caterpillar. Recall that both the body and leg nodes can store an integer.
- You can assume any other method that you might require, but please explain clearly what the method does.

(5)

Q3.

(a)

- Prove that $n \log^3 n$ is $O(n^2)$.
- Prove that n^3 is not $O(n^2 \log n)$.

(5)

(b) The `multiPop(i)` method pops `i` items from the top of a stack. Analyse the amortized complexity of the `multiPop(i)` method.

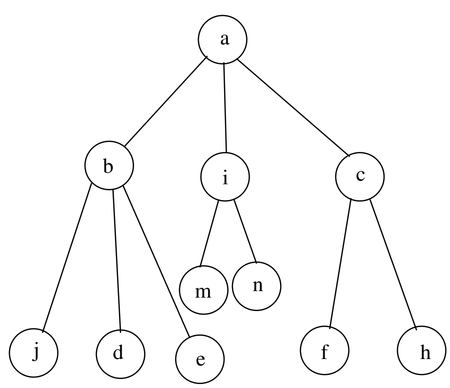
(5)

Q4.

(a) Write pseudocode or explain in your own words the non-recursive depth-first search algorithm for a tree. What is the asymptotic complexity of this algorithm in terms of the number of vertices V of the tree?

(5)

(b) Show the execution of your algorithm in part **(a)** on the tree given below. Show the state of the stack clearly at every step.



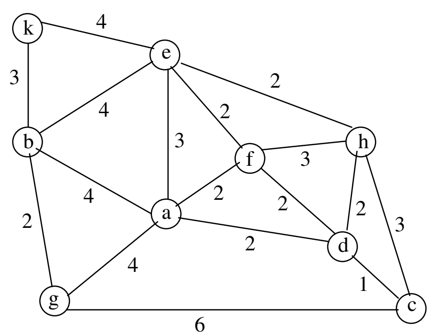
(5)

Q5.

(a) Write pseudocode or explain in your own words Dijkstra's single-source shortest path algorithm. Given a graph $G = (V, E)$, what is the asymptotic complexity of Dijkstra's single-source shortest path algorithm? Explain your answer clearly.

(5)

(b) Show the execution of your algorithm from part **(a)** on the following graph with vertex `a` as the source. The node labels of the graph are shown inside the nodes and the edge weights are shown on the edges.



(5)

———————————–END OF PAPER———————————–