A37232

Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers

UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year Undergraduate

06-30175

LC Data Structure & Algorithms

Main Summer Examinations 2025

Time allowed: 2 hours

[Answer all questions]

Turn Over

Note

Answer ALL questions.

Question 1 Quicksort Variations

Consider the following partition function:

```
int partition(int[] array, int left, int right) {
    int pivot = array[right];
    int[] smaller = new int[array.length]; smallindex = 0;
    int[] bigger = new int[array.length]; bigindex = 0;
    for (int i = left; i < right; i++) {
        if (array[i] < pivot) {
            smaller[smallindex] = array[i];
            smallindex++;
        } else {
            bigger[bigindex] = array[i];
            bigindex++;
        }
    }
    for (int i = 0; i < smallindex; i++)
        array[left+i] = smaller[i];
    array[left+smallindex] = pivot;
    for (int i = 0; i < bigindex; i++)
        array[left+smallindex+1+i] = bigger[i];
    return left+smallindex;
}
```

(a) Using the template below write a Java implementation of (a version of) the Quicksort algorithm that uses the above partition function as a subprocedure. **[5 marks]**

```
void quickSort(int[] array) { this(array, 0, array.length-1); }
void quickSort(int[] array, int left, int right) {
    // your code here
}
```

(b) (i) Is the resulting sorting algorithm stable? Explain your answer. **[4 marks]**

(ii) State the worst case complexity of the algorithm. Describe inputs on which the worst case is achieved, and give an example of such an input for $n = 7$. **[6 marks]**

(iii) State the best case complexity of the algorithm. Describe inputs on which the best case is achieved, and give an example of such an input for $n = 7$. **[6 marks]**

(c) Consider the following random procedure to generate an array: Start with an array `a` of length n in strictly increasing order (i.e., `a[i] < a[i+1]` for all `i < n-1`), choose a position `i = 0, ..., n-1` at random (with all options equally likely), and rotate the array, so that `a[i]` ends up in the last position, i.e., the array will start with `a[(i+1) % n]` followed by `a[(i+2) % n]`, etc. In the typical case, the input will be

```
{a[i+1], a[i+2], ..., a[n-1], a[0], ..., a[i]}
```

Derive the average time complexity of the function `quickSort` on these inputs. Express the result using Θ notation. **[12 marks]**

Hint: Execute the `quickSort` *function on the array* $\{3, 4, 5, 1, 2\}$ *to understand how it behaves.*

Question 2 Mystery Data Structure

The following function non-recursively searches for a value in a tree of integers. It uses an unknown class '`DataType`', with methods '`put`' and '`take`', to temporarily store nodes.

```
class TreeNode {
  int value;
  TreeNode[] children;
}

boolean searchTreeFor(TreeNode root, int value) {
    if (root == null) return false;

    DataType toSearch = new DataType();
    toSearch.put(root);

    while (!toSearch.isEmpty()) {
        TreeNode node = toSearch.take();
        if (node.value == value) return true;
        int n = node.children.length;
        for (int i = 0; i < n; i++) {
            toSearch.put(node.children[i]);
        }
    }

    return false;
}
```

You may assume that each `TreeNode` contains an array of children nodes `children`, which can be empty.

(a) Explain what the terms Breadth-First Search (BFS) and Depth-First-Search (DFS) mean in this context. **[4 marks]**

(b) Assuming that the function is using a BFS strategy:

(i) Using words and diagrams explain what '`DataType`' and its methods '`put`' and '`take`' should be in order to make this algorithm correct. **[5 marks]**

(ii) Give an efficient implementation of this type and these methods in Java code. Do not use any in-built classes except for '`TreeNode`'. **[10 marks]**

(c) Assuming that the function is using a DFS strategy:

(i) Using words and diagrams explain what '`DataType`' and its methods '`put`' and '`take`' should be in order to make this algorithm correct. **[5 marks]**

(ii) Give an efficient implementation of this type and these methods in Java code. Do not use any in-built classes except for '`TreeNode`'. **[10 marks]**

Question 3 A Day in Life of a City Planner

A city's public transport system is modelled as a graph where:

- Each bus stop is a vertex.
- Each direct bus route between stops is an edge.
- The main transport hub is at stop A.

As a planning engineer, you have access to the matrix S whose rows and columns denote different bus stops and $S_{i,j} = 1$ if stop j can be reached from stop i by bus in exactly one stop (i.e., there is a direct bus route from stop i to j), otherwise $S_{i,j} = 0$. You can assume that there are n bus stops in total and m direct bus routes between various pairs of stops.

The city's zoning policy requires each bus stop to be labelled with a zone number based on its shortest distance, in number of bus stops, from stop A (assuming that every stop can be eventually reached from stop A by bus). Specifically:

- The stop A is in zone 0.
- A stop is in zone k if the shortest route to it from stop A has exactly k bus stops.

(a) Describe an algorithm to label each stop with its correct zone number. What is the time complexity of your approach? **[9 marks]**

(b) Assume we are given the following bus routes:

$$A \to B, A \to D, B \to C, B \to D, C \to G, D \to C, D \to E, E \to F, F \to G.$$

(i) Compute the zone labels for all stops. **[5 marks]**

(ii) Suppose a new direct bus route is added between stop B (zone 1) and stop G (zone 3). Explain why it would violate the current zone labelling scheme. **[5 marks]**

(iii) The road planning committee has informed you that the bus route between stop B (zone 1) and stop C (zone 2) is permanently discontinued. However, stop C will remain in zone 2. Show that this will not violate the current zoning policy by giving an example of route (with bus stops and connections) from A to C. **[5 marks]**

(c) Your zone labelling scheme only considered the number of stops and not the distance the bus travels. As the fuel cost (and therefore the ticket price) depends on the actual distance, the new government wants to change the zoning policy. The new zoning policy states that:

- The stop A is in zone 0.
- A stop is in zone k if the distance a bus travels to cover the shortest route to it from stop A is exactly k.

You have access to the distances between stops for all bus routes as before, in terms of the matrix S, but now $S_{i,j}$ shows the distance between stop i and j if there is a direct bus route from stop i to j, and zero otherwise. Describe an algorithm that label each stop with its correct zone number. What is the time complexity of your approach? **[9 marks]**

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.