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

Data Structures and Algorithms

Main Summer Examinations 2024

Time allowed: 2 hours

[Answer all questions]

Turn Over

Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Question 1

(a) Consider an integer array containing the values 4, 8, 6, 3, 7, in that order. Draw the state of the array after each iteration of the outer loop of the following algorithms. For each such array state, mark the point that separates the sorted part of the array from the unsorted part.

(i) Insertion Sort **[4 marks]**

(ii) Selection Sort **[4 marks]**

(b) Explain what is meant by saying a sort algorithm is *stable*. **[2 marks]**

(c) Is Insertion Sort stable? Justify your answer. **[4 marks]**

(d) Is Selection Sort stable? Justify your answer. **[4 marks]**

(e) Aside from stability, are their any other factors that would recommend choosing one of these two algorithms over the other? **[2 marks]**

Question 2

A non-circular, singly linked list Abstract Data Type has a private `Node` pointer, called `first`, which points to the first node of the list. This ADT does not maintain a header pointer to the last node of the list, but it does maintain a variable `size` to keep track of the number of nodes in the list.

Given a pointer to a node `n`, assume that values can be read or written to by using the field names `n.val` and `n.next` for the value and next fields of the node respectively.

Assume that a garbage collector is being used, so nodes do not need to be explicitly freed. Use `END` for invalid or *null* address pointer values.

(a) `delete_nth(int n)` is a **non-recursive** method of the ADT that removes node number *n* from the list and throws an `IllegalArgumentException` if no such element exists or if *n* is less than 0. The first node is numbered 0.

Fill in the missing part of the pseudocode for this function below:

```
1  void delete_nth(int n)
2  {
3    // WRITE THE CODE THAT SHOULD BE HERE
4    return
5  }
```

[10 marks]

(b) Assume that this ADT has two different different methods to delete all the elements from the singly linked list as follows:

```
1  void delete_all_from_start()
2  {
3    while (size > 0)
4      delete_nth(0)
5    return
6  }
7
8  void delete_all_from_end()
9  {
10   while (size > 0)
11     delete_nth(size - 1)
12   return
13 }
```

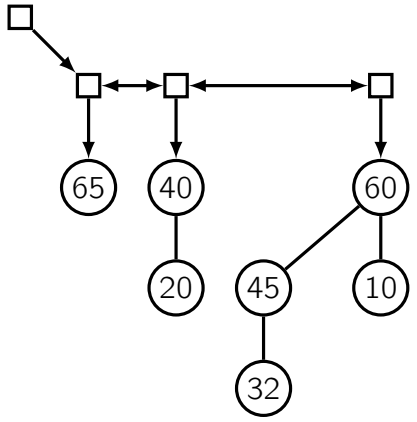
(i) Give the complexities of these two methods in big O notation and explain how you derived them.

(ii) Write the pseudocode for a much simpler and more efficient method to delete all elements from the list.

[10 marks]

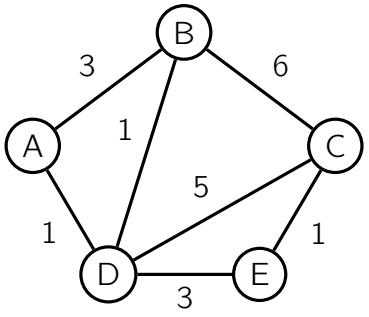
Question 3

(a) Draw a diagram of the result of inserting the value **55** into the following Binomial Heap, where the highest priority is the highest integer value, and explain the steps by which you came to your result.



[10 marks]

(b) Consider the following graph:



Demonstrate the execution of Dijkstra's Algorithm for finding the minimal path on this graph from node A to node C, by writing out a table of the execution steps. The first row of these steps, following initialisation, is already provided. The remaining steps should be written in the same format:

A	B	C	D	E	Finished
0,A	∞, B	∞, C	∞, D	∞, E	
⋮	⋮	⋮	⋮	⋮	⋮

Each row should show the results of one iteration of the algorithm, where the *Finished* column identifies the node that is finished in that iteration, and the remaining columns show the current shortest distance of the node of that column from node A, the previous node to the node of this column on the current shortest path from A and a tick mark if the node of the column is finished. Finally, explain how the shortest path, and its length, can be read from the final line of the table on completion of the algorithm. **[10 marks]**

End of Paper

This page intentionally left blank.

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.