

**OLLSCOIL NA hÉIREANN, CORCAIGH**  
THE NATIONAL UNIVERSITY OF IRELAND, CORK

**COLAISTE NA hOLLSCOILE, CORCAIGH**  
UNIVERSITY COLLEGE, CORK

2016/17

Semester 1 - Winter 2016

**CS2515 Algorithms and Data Structures I**

Dr Helen Purchase  
Professor Cormac Sreenan  
Professor Ken Brown

1.5 Hours

The use of electronic calculators is permitted

Answer all questions

Total Marks: **80**

(~1 minute per mark)

**PLEASE DO NOT TURN THIS PAGE UNTIL  
INSTRUCTED TO DO SO**

**PLEASE  
ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER**

1. (This complete question is worth 30 marks)

- (i) Explain the implementation of a *Doubly-Linked List* using Python classes and objects, giving a clear specification for each class, object and variable. You do not need to specify any methods. Include in your explanation an illustration of how the sequence of objects  $\langle x, y, z \rangle$  would be represented.

(7 marks)

- (ii) Give clear pseudocode for the operation of swapping two neighbouring nodes in a doubly linked list, using the implementation you specified in 1(i) above. (Note: your pseudocode must swap the nodes, and not simply swap the data)

(4 marks)

- (iii) State the defining characteristics of a *Stack*, and state the standard operations offered in the Stack ADT. For each operation, give a brief comment explaining what it does.

(5 marks)

- (iv) Explain, using text or sketches of the layout in memory or pseudocode or Python code, how we can use a doubly-linked list to give an efficient implementation of a Stack. State the complexity of each operation.

(6 marks)

- (v) A *MinStack* is an ADT which extends the Stack ADT with one additional method, *min()*, which simply reports the minimum valued element in the stack. Give a design for an efficient implementation of MinStack – ideally, this implementation should allow constant time operation (i.e.  $O(1)$ ) for *min()*, and should not change the complexity of any of the other methods compared to original Stack.

(8 marks)

2. (This complete question is worth 30 marks)

- (i) State the defining characteristics of a *Priority Queue*, and state the standard operations offered in the Priority Queue ADT. For each operation, give a brief comment explaining what it does.

(6 marks)

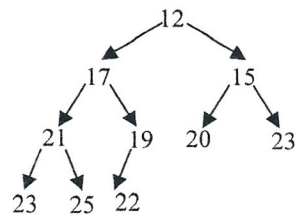
- (ii) Describe how to implement a Priority Queue using an *unsorted* Doubly-linked list, and state the worst-case complexities of the operations.

(5 marks)

- (iii) State the defining characteristics of a (*min*) *Binary Heap*, give a clear description (using text, diagrams or pseudocode) of the operations to add and remove elements, and explain how it can be used to implement a Priority Queue. State the complexity of the add and remove operations for a heap with  $n$  elements.

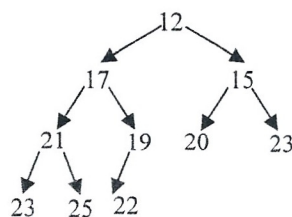
(8 marks)

- (iv) For the min binary heap below, show the resulting binary heap when you add element 14.



(3 marks)

- (v) For the min binary heap below, show the resulting binary heap when you remove element 12.



(3 marks)

- (vi) Explain how a min binary heap can be represented using a Python list, and explain in detail how the add and remove methods can be implemented efficiently. Use the example heap shown in (v) above as an example to support your explanation.

(5 marks)

3. (This complete question is worth 20 marks)

- (i) State the defining characteristics of a *Map* (i.e. a *dictionary* in Python), and state the standard operations offered in the Map ADT. For each operation, give a brief comment explaining what it does.

(6 marks)

- (ii) Explain the concept of a *bucket array*, and outline the basic implementation of a Map using a bucket array.

(5 marks)

- (iii) Explain *Open addressing* and *Linear probing* as a method to implement a Map without requiring a bucket array, yet giving expected constant time complexity for updates and retrieval.

(9 marks)

---

**Total: 80 marks**

**END OF PAPER**