



UNIVERSITY OF MORATUWA

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BSc Engineering Honours Degree
2011 Intake Semester 4 Examination

CS 2022: DATA STRUCTURES AND ALGORITHMS

Time allowed: 2 Hours

April/May 2014

ADDITIONAL MATERIAL: *None*

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of **FOUR (4)** questions in **FIVE (5)** pages.
2. Answer **ALL** QUESTIONS
3. Start answering each main question on a new page.
4. All questions carry equal marks.
5. This examination accounts for 60% of the module assessment.
6. This is a closed book examination.

NB: It is an offence to be in possession of unauthorised material during the examination.

7. Only calculators approved and labelled by the Faculty of Engineering are permitted.
8. Assume reasonable values for any data not given in or with the examination paper. Clearly state such assumptions made on the script.
9. In case of any doubt as to the interpretation of the wording of a question, make suitable assumptions and clearly state them on the script.
10. This paper should be answered only in English.

Continued...

Q1. [25 marks]

Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A .

- List all the inversions in the array [8, 5, 7, 4, 1]. [3 marks]
- Show how the array [8, 5, 7, 4, 1] will be sorted by the bubble sort algorithm. [4 marks]
- What is the relationship between the **number of swaps that will occur in Bubble sort and the number of inversions in the input array**? Explain your answer.. [4 marks]
- Analyze the worst case time complexity of the optimized version of bubble sort is shown below. [7 marks]

OPTIMIZED-BUBBLE-SORT (A)

```

1.  n = A.length
2.  do
3.      swapped = false
4.      for i = 2 to n
5.          if A[i-1] > A[i]
6.              temp = A[i]
7.              A[i] = A[i-1]
8.              A[i-1] = temp
9.              swapped = true
10.         newLimit = i-1
11.         n = newLimit
12. while swapped

```

- Give the asymptotic growth in “big oh” notation for the following functions. Please show how you obtained your answer.
 - $T(n) = 10^{20}n + (3 \times 10^{-10})n^2 + 0.9n \log n$ [3 marks]
 - $T(n) = 3T\left(\frac{n}{2}\right) + n \log n$ [4 marks]

Q2. [25 marks]

- What is an abstract data type (ADT)? [4 marks]
- Compare and contrast array based implementation and the single linked implementation approaches that can be used to represent/implement K-ary trees. **You should consider the space efficiency, and time complexity of accessing elements in your comparison.** [4 marks]

Continued...

- c) Explain a suitable representation (data structure) that can be used to implement trees with unbounded branching. [4 marks]
- d) Write pseudo code to implement a tree with unbounded branching with the following operations using the representation you explained in part (c). The **tree should store an integer value in each node and for a given node, the child nodes are ordered in the ascending order** (i.e. the child nodes of each node are sorted in ascending order).
- Given the parent node and an integer, add the new value as a child node of the parent. (Note: You have to make sure that the sorted order of the child nodes is maintained) [4 marks]
 - Given a parent node and an integer, delete that number if it is present as a child node of that node. In case the node being deleted has child nodes, they should not be deleted and should be attached as child nodes of the parent node of the node that is being deleted. [9 marks]

Q3. [25 marks]

- a) Consider the integer array {4, 1, 3, 2, 16, 9, 10, 14, 8, 7}, which represents a binary tree.
- Draw the binary tree that is represented by the array. [2 marks]
 - Show how **Build-Max-Heap** operation will transform the above binary tree into a max heap. (You can show your steps using array representation or tree representation of heap.). [4 marks]
 - Show how Heapsort can be used to get the sorted list of numbers in the heap built in part (ii). (You can show your steps using array representation or tree representation of heap.) [5 marks]
- b) Assume you are given an implementation of a heap data structure which has the following operations.
- `int Parent(int i)` - This function will return the index of the parent node of the node *i*.
 - `int Left(int i)` - This function will return the index of the left child node of the node *i*.
 - `int Right(int i)` - This function will return the index of the right child node of the node *i*.
 - `void Max-Heapify(int[] A, int i)` - Given a node *i*, with left child *l* and a right child *r* and with both the sub -trees rooted by *l* and *r* are max-heaps, this function will convert the sub-tree rooted by node *i* a max-heap.
 - `void BuildMax-Heap(int[] A)` - This function will convert the specified array into a max-heap.

Continued...

- `element_t HeapMaximum(int[] A)` - This function will return the maximum element in the heap.
- `element_t HeapExtractMax(int[] A)` - This function will remove the maximum element from the heap and return it.
- `int Max-HeapInsert(int[] A, int x)` - This function will insert the specified element to the max heap.

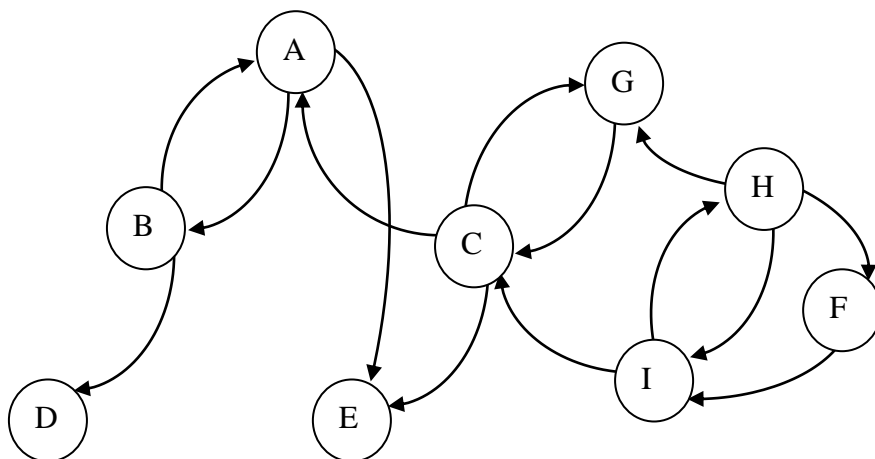
Write Pseudocode to implement a Max Priority Queue using the above implementation. Your priority queue should support following operations.

- Initializing the priority queue. [2 marks]
- Inserting an element to the priority queue. [3 marks]
- Removing the item with highest priority from the queue. [4 marks]
- Increasing the value of a specified item in the priority queue. [5 marks]

Q4. [25 marks]

- a) Generate the Breadth First Search Tree for the following graph starting from vertex C.

[5 marks]



Continued...

b) Bellman-Ford algorithm and Dijkstra's Algorithm are two algorithms for calculating Single Source Shortest Path (SSSP).

- i. Explain how a SSSP algorithm can be used to solve the single destination shortest path problem in which you are expected to calculate the shortest path to a specified vertex starting from each of other vertices. [3 marks]
- ii. Using one of the above algorithms, calculate the shortest path to A from all other vertices in the graph shown in following diagram (Figure Q4). Clearly indicate all the steps you followed. [7 marks]

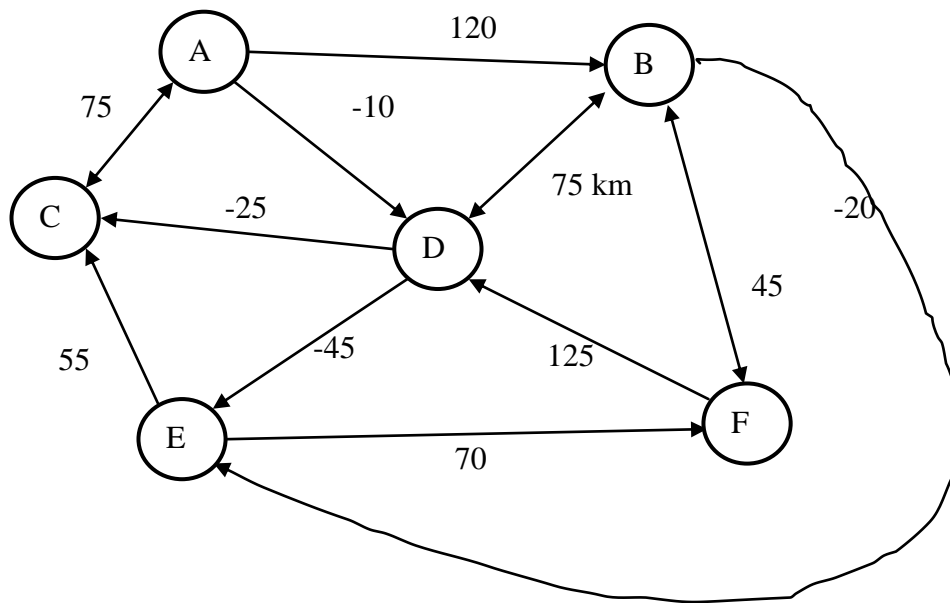


Figure Q4

- c) Dynamic Programming is a technique which lets you solve certain set of problems efficiently compared to recursion. In order to take advantage by applying the dynamic programming technique, the problem should have some properties. Briefly explain each of those properties. Using those properties, explain why dynamic programming is more efficient compared to brute force technique. [6 marks]
- d) What is greedy choice property? Briefly explain why greedy solutions would run faster compared to dynamic programming solutions. [4 marks]

--- End of the Paper ---