UNIVERSITY OF MORATUWA

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BSc Engineering Honours Degree
Semester 2 Examination: 2010/2011

CS2022: DATA STRUCTURES AND ALGORITHMS

Time allowed: 2 Hours December 2011

ADDITIONAL MATERIAL: *None*

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of **4** questions in 5 pages.
2. Answer all **4** questions.
3. Start answering each of the 4 main questions on a new page.
4. The maximum attainable mark for each question is given in brackets.
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.

Continue...

Question 1

(a) What is an algorithm? List four (4) factors that should be considered in evaluating an algorithm. [5]

(b) You are provided with an implementation of a (single) linked list which supports the following operations.

- *LIST-SEARCH*(L,k): Finds the first node with the key k in the list L. Returns *NIL* if no element is found.
- *LIST-INSERT*(L,x): Inserts the value x into the front of the list L.
- *LIST-DELETE*(L,x): Deletes the first node which contains value x from the list L if it exists.
- *LIST-INSERTAT*(L,x,i): Inserts the value x into the i^{th} location of the list L.
- *LIST-DELETEAT*(L,i): Deletes the node in the i^{th} location of the list L.

Using this implementation as a library (you are not allowed to change any operations of the provided linked list implementation), write the pseudo code to implement the following abstract data type (ADT) specifications.

i. ADT Queue which has enqueue and dequeue operations. [8]

ii. ADT Stack which has push and pop operations. [8]

(c) If you use a double linked list instead of a single liked list for the implementation of the ADT specifications of part (b) above, will there be any advantage? Please explain. [4]

Question 2

(a) A heap is a special type of binary tree. What are the properties of a **min-heap**? [2]

(b) Show, using a suitable diagram, a **min-heap** formed by the following numbers:
22 6 4 2 55 35 46 35 35 44 62 12.
Clearly show the steps you followed. [7]

(c) What is a hash table? Explain the insert and search operations of a hash table. Mention the average case and worst case time complexities for the two operations and explain how these time complexities are achieved by hash tables. [5]

(d) **Briefly** explain three (3) collision handling mechanisms used in hash tables. [6]

(e) Give the asymptotic growth in "big oh" notation for the following functions. ***Please show how you obtained your answer***.

i. $T(n) = 1222\, n + (3\times10\text{-}22)\, n^2 + 0.9n \log n$ [2]

ii. $T(n) = T(\lfloor\sqrt{n}\rfloor) + n$ [3]

Continue...

Question 3

(a) Adjacency matrix representation and adjacency list representation are two popular graph representation mechanisms.

i. **Briefly** explain how the above mentioned two mechanisms represent graphs. [4]

ii. Compare above two representations considering the space efficiency, ease of adding a new vertex to the graph, and the time required to check the presence of an edge in the graph. [3]

(b)

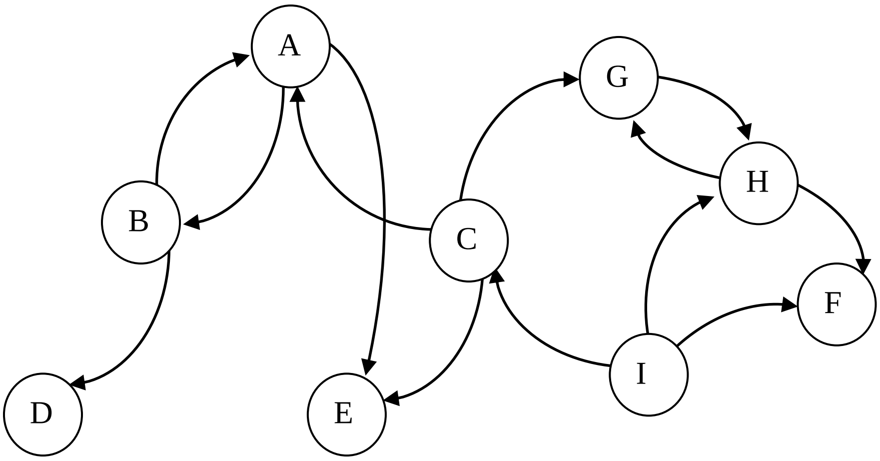

Figure Q3_1

i. Generate a depth first forest for the graph shown above (Figure Q3_1). ***Clearly show the steps you followed.*** [6]

ii. Identify the type of each edge in the graph (in Figure Q3_1) (whether a tree edge, cross edge, back edge or a forward edge). [2]

(c)

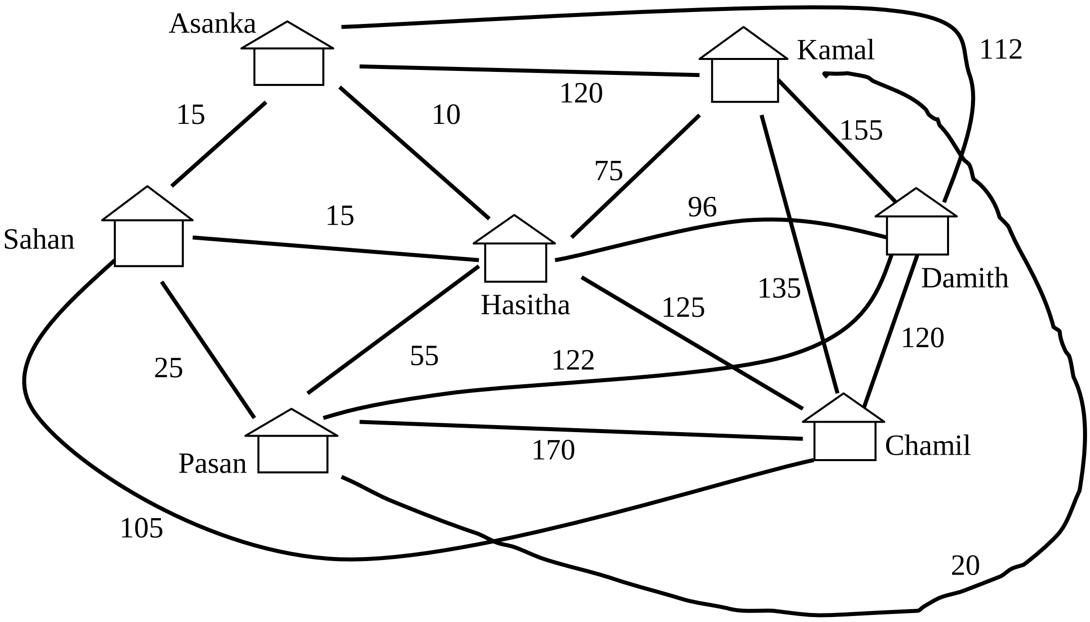

Figure Q3_2: Homes and the Cost (in Rs. 1000s) of Connecting the Homes

Continue...

Above diagram (Figure Q3_2) shows the homes of seven friends (Asanka, Chamil, Damith, Hasitha, Kamal, Pasan and Sahan) and the cost of connecting their homes through bi-directional communication links. Note that it is not possible to connect certain pairs of houses using a direct communication link. The friends are planning to connect their houses to into a single network with a minimum connection cost. Damith proposes that they can select a set of connections that will minimize the cost using a minimum spanning tree generation algorithm.

i. Name a minimum spanning tree generation algorithm that can be used to select a set of connections that will minimize the connection cost for the above scenario. [1]

ii. Explain the **basic idea** (you do not have to specify the pseudo code or the detailed algorithm.) of a minimum spanning tree generation algorithm you specified as answer to **above** part (i). [3]

iii. Calculate a set of connections that will minimize the connection cost using the algorithm you specified in part (i). ***Clearly show the steps you followed in obtaining the answer.*** [6]

Question 4

(a) Explain the steps required in deleting a node from a binary search tree. [4]

(b) Show the binary search tree (BST) obtained by inserting the following sequence of numbers into an empty BST; 82, 67, 98, 72, 80, 70, 12, 25, 84, 108, 122, 135, 120. ***Clearly show the steps followed.*** [3]

(c) Take the binary search tree obtained as the answer of part (b) and remove the nodes with values 120, 12 and 67 in the same order. Show the new BST created after these operations. ***Clearly show the steps followed.*** [3]

(d) 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. [4]

(e) What is the greedy-choice property? [2]

Continue...

(f) The pseudo code for the merge sort algorithm is shown below.

```
MERGE-SORT(A, p, r)
   IF p < r
         q ← ⌊(p + r)/2⌋
         MERGE-SORT(A, p, q)
         MERGE-SORT(A, q + 1, r)
         MERGE(A, p, q, r)

MERGE(A, p, q, r)
   n1 ← q - p + 1
   n2 ← r - q
   //create arrays L[1………n1 + 1] and
   R[1  n2 + 1]
   for i ← 1 to n1
         L[i] ← A[p + i - 1]
   for j ← 1 to n2
         R[j] ← A[q + j]
   L[n1 + 1] ← ∞
   R[n2 + 1] ← ∞
         i ← 1
         j ← 1
         for k ← p to r
               if L[i] ≤ R[j]
                     A[k] ← L[i]
                     i ← i + 1
               else
                     A[k] ← R[j]
                     j ← j + 1
```

Using this pseudo code, show that the running time of the Merge sort is O(***n*** *log* ***n***) [9]

========== END OF THE PAPER ==========