COMP 2119A Introduction to Data Structures and Algorithms
Assignment Two
Due Date: 25 October 2016 midnight

Assignment box: A3 (also submit a scanned copy to moodle)

[Note that for the questions involving algorithm design, before you present your algorithm, try to describe your idea first!
More marks will be given to faster algorithms.]

** NO LATE ASSIGNMENT will be accepted **

1) [10%] Design a <u>recursive</u> algorithm to solve each of the following problems and analyze the time complexity of your algorithm.
   (a) Given an array of n numbers, reverse the entries in the array and you are only allowed to use constant amount of extra storage (i.e., you cannot use an extra array.).
   (b) Given a <u>*sorted*</u> array of *distinct* integers $A[1..n]$ in increasing order, find out whether there is an index i for which $A[i] = i$.

2) [10%] Given an array $A[1..n]$ of n integers representing the stock price of n days, the problem is to find i, j such that $A[j] - A[i]$ gives us the maximum over all possible i and j (i.e., if we buy at i-th day and sell at j-th day, we earn the most). Your algorithm should also report the maximum value of $A[j] - A[i]$.
   (a) Present the brute-force approach (i.e., trying all possible i, j and report the maximum of $A[j] - A[i]$) and analyze the time complexity of your algorithm.
   (b) Design a recursive algorithm to solve the problem and analyze the time complexity of your algorithm.

3) [15%] Consider the following variation of Tower of Hanoi. There are 4 pegs (0 to 3). A pile of n red disks with increasing sizes, is initially on peg 0, with the largest disk at the bottom. Another pile of n blue disks is initially on peg 3. Except the color, these two piles are identical in sizes. We want to move all red disks to peg 3 and all blue disks to peg 0. Initially, pegs 1 and 2 are empty. The same set of rules as the original Tower of Hanoi applies: (i) Each time we can move only one disk; (ii) A disk cannot be put on top of another disk with a smaller size. Design an algorithm to solve the problem and analyze the time complexity of your algorithm.

4) [10%] Consider the ADT stack. In addition to the operations, **Push**, **Pop**, **Top**, we have to support a new operation, **FindMin**, which returns the smallest element in the stack. Design a data structure to support these operations such that each operation takes constant time. Analyze the time complexity of these four operations. [No need to check the overflow and underflow conditions and no need to give the procedures for **Empty** and **Full**. [Hint: use an extra stack.]

5) [10%] Describe how you can represent a n-vertex undirected pseudograph using an $n \times n$ adjacency matrix. And present algorithms and analyze their complexities for answering the following questions.

(a) Given a vertex i, degree of i. (A self loop counts 2 to the degree of the vertex.)
(b) Given two vertices i and j, report if j can be reached from i in the graph.
(c) Determine if a given undirected pseudograph is connected or not.
(d) Report if the graph has a cycle (self loop does not count).

6) [10%] Given an undirected graph G and a starting vertex s in G, we want to enhance the breadth-first search algorithm and write an additional function print(v) to print a shortest path from s to any vertex v.

(a) First, we enhance the breadth-first search algorithm as follows. For each vertex v_j, use an array PREV() to store the vertex v_i where v_i is the second last vertex of the shortest path from s to v_j. Show this enhanced algorithm. [No need to prove the correctness of your algorithm.]
(b) Design a recursive algorithm print(v) that makes use of the array PREV() to print a shortest path from s to the vertex v. [You can assume that PREV() is global.]

7) [10%] Four people (A to D) come to a river at night. There is a narrow bridge that can only hold two people at the same time. They have only one torch. They need to use the torch when crossing the bridge as it is completely dark. Person A can cross the bridge in w minutes, B in x minutes, C in y minutes, and D in z minutes. When two people cross the bridge together, they must move at the slower person's pace. The problem is to figure out a way to let them cross the river using the shortest amount of time.

Model this problem as a graph problem. State clearly (i) what type of graph you will use; (ii) what a node/vertex represents; (iii) what an edge represents; and (iv) the corresponding graph problem to be solved.
[Hint: Think about how to represent the time required for two people to cross the bridge in the graph – weighted graph?]

8) [10%] (MIT Q11.4-1, p.244)
Consider inserting the keys 10, 22, 31, 4, 15, 28, 17, 88, 59 into an (initially empty) hash table of length $m = 11$ using open addressing with primary hash function $h'(k) = k \bmod m$. Illustrate the result of inserting these keys using (i) linear probing, (ii) using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and (iii) using double hashing with $h_2(k) = 1 + (k \bmod (m - 1))$. [You do NOT need to show the intermediate steps.]

9) [10%] Consider the following hash function: $h(k, i) = (h'(k) + \frac{1}{2}(i + i^2)) \bmod m$, where $m = 2^p$ for some positive integer p. Prove or disprove that for any k, the

probe sequence <h(k, 0), h(k, 1), ..., h(k, m – 1)> is a permutation of <0, 1, 2, ..., m – 1>.

10) [5%]

(a) Is the assignment (1) too difficult; (2) too easy; (3) about right?

(b) How many hours you spend on the assignment?

   (1) Less than 5 hours
   (2) 5 – 10 hours
   (3) 10 – 20 hours
   (4) More than 20 hours

(c) [Self assessment] Do you consider yourself understand the topics of this assignment?

   (1) Yes; (2) not 100% sure; (3) No

   If your answer is (2) or (3), please elaborate (at least indicate which part you do not understand).

(d) Other comments?

--- End of Assignment ---