**CS3230: Design and Analysis of Algorithms (Fall 2014)**

**Tutorial Set #10**

[For discussion during Week 12]

**S-Problems are due (outside Prof. Leong's office): Friday, 31-Oct, before noon.**

**OUT:** 28-Oct-2014 **Tutorials:** Tue & Wed, 4, 5 Nov 2014

**IMPORTANT: Read "Remarks about Homework".**

**Submit solutions to S-Problem(s) by deadline given above.**
**Prepare your answers to all the D-Problems in every tutorial set**.

When preparing to present your answers,

- Think of a CLEAR EXPLANATION
- Illustrate with a good worked example;
- Describe the main ideas,
- Can you sketch why the solution works;
- Give analysis of running time, if appropriate
- Can you think of other (perhaps simpler) solutions?

**Helpful Hints Series:**
**Please note that getting the right recursion is quite important for dynamic programming algorithms.**
**Similarly proof of correctness is quite important for greedy algorithms.**

***Routine Practice Problems*** *-- do not turn these in -- but make sure you know how to do them.*

**R1.** What is the data structure used by Prim's algorithm for efficient implementation?

**R2.** What is the data structure used by Kruskal's algorithm for efficient implementation?

**R2.** Bellman-Ford algorithm works well with real edge weights? True or False?

**R3**. Bellman-Ford algorithm can be used to produce shortest paths between all pairs of vertices? True or False?

**S-Problems: (To do and submit by due date given in page 1)**
Solve this S-problem(s) and submit for grading.

**IMPORTANT: Write your NAME, Matric No, Tutorial Group in your Answer Sheet.**

### S1. [Longest increasing subsequence]

Write a dynamic-programming algorithm (idea and pseudocode) to find the length of a longest increasing subsequence of a given sequence of numbers. For example in the sequence 3,5,4,9; a largest increasing subsequence is 3,4,9, another is 3,5,9 (note however that both have the same length 3).

(Hint: One approach could be to use what you did in **D3**.)

**D-Problems:** Solve these D-problems and prepare to discuss them in tutorial class. You may be called upon to present your solution *or your best attempt at a solution*. Your solution presentation does NOT need to be fully correct, given your best attempt. The TA will help clarify and correct any issues or errors.

### D1. [Unique MST]

Prove that a directed weighted graph with distinct edge weights has a unique minimum spanning tree (MST).

### D2. [Reverse Delete Algorithm]

Prove the correctness of the Reverse Delete Algorithm for Minimum Spanning Tree problem (assume distinct edge weights and hence a unique MST for the graph).

### D3. [Longest common subsequence]

A string s is a subsequence of another string t, if it can be obtained by deleting some characters in t. For example a longest common subsequence of `a b c d e f' and `c d a b c f' is `a b c f'.
Write a dynamic-programming algorithm (idea and pseudocode) to determine the length of a longest common subsequence of two given strings.

**D4. [Longest common substring]**

A substring is a subsequence where all the items are consecutive. For example a longest common substring of `a b c d e f' and `c d a b c f' is `a b c' (please note the difference with longest common subsequence).

Write a dynamic-programming algorithm (idea and pseudocode) to determine the length of a longest common substring of two given strings.