# CSI 3105 ASSIGNMENT 4 Fall 2014

## (LAST ONE! ☺ )

**Due Date:** Monday, December 1, 2014, by 5:30 pm. You can hand in your assignment directly to me in class, or else you can bring it to my office (STE 5106) before the due time (put it under my door if I am not there). Assignments from 1 minute to 24 hours late will lose 6 out of 60 marks (10%). Assignments received after 5:30 pm on December 2 will receive a grade of 0.

**On this and all future assignments**: You must be sure to include your name and student ID on the assignment, as well as the course code and the assignment number. In your assignment, please be sure to explain all of your solutions CLEARLY, or you will lose marks. To be complete, in any worst case analysis you do you must state the input size, the basic operations being counted, and a general input of your fixed input size which gives the worst case (you should state these things in your solution even if some of them are given in the questions). Whenever possible, simplify your answers by using any appropriate formulas found in Appendix A. Also, you will be marked on the efficiency of any algorithms you design.

This assignment will be marked out of 60.

---

1. [10 marks]
Describe how the MST algorithm in the text (Algorithm 4.1, which is the implementation of Prim's algorithm) could be modified so that it would solve the following problem: Given an undirected weighted graph G, check to see if G is connected. Note: Assume G is given in the same adjacency matrix format that is used by Algorithm 4.1.

2. [10 marks]
Use the greedy algorithm from class to find a stable configuration for the stable marriage problem given below. You do not need to show your individual steps, just the final solution. You may choose which group does the asking (the women or the men) but please be sure to STATE which group is doing the asking for your solution.

| | Women | Men | |
|---|---|---|---|
| (5, 1, 3, 4, 2) | A | 1 | (A, D, B, C, E) |
| (2, 4, 3, 5, 1) | B | 2 | (E, B, A, C, D) |
| (3, 5, 1, 4, 2) | C | 3 | (D, B, C, E, A) |
| (3, 2, 4, 5, 1) | D | 4 | (B, D, A, E, C) |
| (3, 2, 1, 5, 4) | E | 5 | (B, C, E, A, D) |

3. [10 marks]
Using the greedy method TAUGHT IN CLASS (which is different than the one in the text), find a maximum profit schedule for the following scheduling-with-deadlines problem:

| Job | Deadline | Profit |
|---|---|---|
| 1 | 4 | 40 |
| 2 | 4 | 15 |
| 3 | 3 | 60 |
| 4 | 1 | 20 |
| 5 | 3 | 10 |
| 6 | 1 | 45 |
| 7 | 1 | 55 |
| 8 | 3 | 15 |

Please show your work--more specifically, for each stage of the algorithm state which job you are considering next, and the time slot in which it gets scheduled by the algorithm, or mention it doesn't get scheduled (but you don't have to explain why). Be sure, at the end, to state the final answer, i.e. state the final schedule, and the total profit it gives.

4. [10 marks total]

a) [8 marks]

Draw the decision tree for n=3 for the Mergesort algorithm, which is Algorithm 2.2 in your text. For your decision tree, let S = (a, b, c). Note: Be sure to mark the branches of the tree as Yes and No, where the Yes branches are always to the left.

b) [1 mark] From the tree, what is W(3) for Mergesort? Briefly explain your answer.

c) [1 mark] From the tree, what is the fewest comparisons done by Mergesort for any input of size 3? Again, please briefly explain your answer.

5) [20 marks total]

Recall that for the CNF satisfiability problem (SAT, for short), that the problem is called k-SAT if every clause in the expression has exactly k literals per clause.

a) [18 marks]

Suppose we wish to show 4-SAT is NP-complete. This requires us to show that 4-SAT is in NP, and also that B $\propto$ 4-SAT for some problem B that is known to be NP-complete.

Suppose we already know that 4-SAT is in NP, and that 3-SAT is NP-complete. Complete a proof that 4-SAT is also NP-complete by showing 3-SAT $\propto$ 4-SAT, i.e. describe clearly a polynomial-time transformation TRAN that takes an instance x of 3-SAT and reduces it to an instance y of 4-SAT (i.e. y = TRAN( x ) is an instance of 4-SAT). You will need to prove the following three things about TRAN:

i) TRAN is a polynomial-time algorithm.

ii) A yes answer to y = TRAN ( x ) implies a yes answer for an instance x of 3-SAT.

iii) A yes answer to an instance x of 3-SAT implies a yes answer for y = TRAN( x ).

b) [2 marks]

Demonstrate your transformation TRAN on the following instance of 3-SAT (i.e. show what you get for the transformed problem):

$$(\bar{x}_1 \vee x_2 \vee \bar{x}_4) \quad \wedge \quad (x_1 \vee \bar{x}_2 \vee x_3)$$

NOTE: The following type of solution for TRAN is INCORRECT:

Take the instance of 3-SAT, and add a “F” (false) literal to every clause.

i.e. the example in b) would become

$$(\bar{x}_1 \vee x_2 \vee \bar{x}_4 \vee F) \quad \wedge \quad (x_1 \vee \bar{x}_2 \vee x_3 \vee F)$$

It is true that the answer is yes for x if and only if the answer is yes for y here, the problem is that the transformed problem y is NOT an proper input instance of 4-SAT. The input for a k-SAT problem does not include setting a particular variable value to true or false.