CSI 5165 Combinatorial Algorithms — Fall 2015
Computer Science — University of Ottawa

Homework Assignment #3 (100 points, weight 8%)
Due: to be decided.

---

1. (45 points) **Heuristic searches for Maximum cliques** (written question)
   Develop a hill-climbing algorithm, a simulated annealing algorithm and tabu search algorithm for the Maximum Clique problem (the problem of finding a clique of maximum cardinality in a graph).

   Before giving the **pseudocode** for each algorithm, describe your choices for neighbourhood function. If some of these features are common to more than one algorithm, please explain them only once, indicating in which algorithms they will be used.

   For each algorithm, write a paragraph explaining which parameter variations you recommend to be tried in order to experiment with each algorithm.

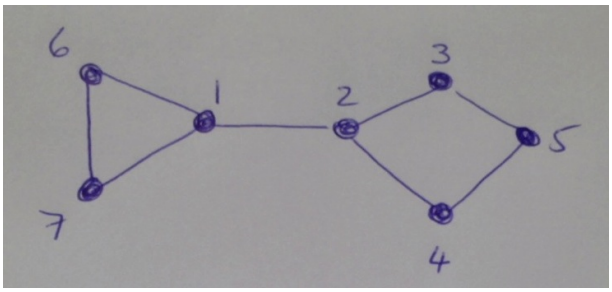
   You will be marked for clarity, conciseness, and quality of algorithm design.

2. (10 points) **Finding isomorphism by hand**
   Do exercise 7.1 of the textbook.

3. (10 points) **Certificate for trees**
   Do exercise 7.2 of the texbook. Simulate the algorithm by hand computation, showing your tree and labels at each step.

4. (10 points) **Reverse the certificate for a tree**
   Do exercise 7.3 of the textbook. Show how the tree is built step by step.

5. (25 points) **Certificate for graphs**
   Apply Algorithm 7.8 by hand, in order to compute a certificate for the following graph. Show your backtracking tree. (Note that before the first branch is explored, there is a call to procedure Refine).

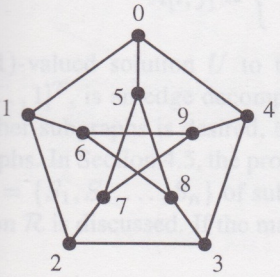


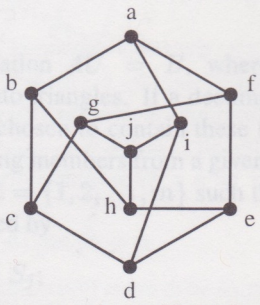
   Note: there are less than 10 nodes in the tree (in case yo make a mistake and get too many subtrees, this note should save you time!).

Exercises of the textbook included.

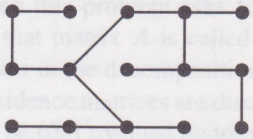
Exercises

7.1 Find an isomorphism between the following two graphs. (These are two different representations of the Petersen graph.)





7.2 Use the algorithm described in Section 7.3.1 to compute the certificate for the tree given below



7.3 Use the algorithm described in Section 7.3.1 to compute the tree whose certificate is 00001011100101011001110000111000111 1.