CSI 4105 Design and Analysis of Algorithms II — Fall 2003
Computer Science — University of Ottawa

Homework Assignment #3 (100 points, weight 10%)
Due: November 27 (10:00 a.m.) in lecture.

---

1. (20 points) **Another vertex-cover greedy heuristic**
Professor Nixon proposes the following heuristic to solve the vertex-cover problem. Repeatedly select a vertex of highest degree and remove all of its incident edges. Give an example to show that the professor's heuristic does not have an approximation ratio of 2.
Hint: try a bipartite graph with vertices of uniform degree on the left and vertices of varing degree on the right. You don't have to use the hint.

---

2. (40 points) **Polytime vertex cover for trees**

1. (20 marks) Give an efficient greedy algorithm that finds an optimal vertex-cover for a tree. It has to be polynomial time and efficiency counts.
2. (10 marks) Prove that your algorithm gives an optimal vertex-cover.
3. (10 marks) Analyse the running time of your algorithm.

Note: the tree is not rooted, it is just a graph with no cycles.

---

3. (20 points) **Set covering approximation algorithm**
Consider each of the following words as a set of letters:
$\{arid, dash, drain, heard, lost, nose, shun, slate, snare, tread\}$.
Show which set cover GREEDY-SET-COVER produces when ties are broken in favour of the word that appears first in the dictionary.

---

4. (20 points) **An euclidean TSP question**
Suppose that the vertices for an instance of the traveling-slaesman problem are points in the plane and that the cost $c(u, v)$ is the euclidean distance between points u and v. Show that an optimal tour never crosses iteself.
Hint: suppose that a tour crosses itself, and prove you can build a tour of cost strictly smaller than it.
