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

Homework Assignment #1 (100 points, weight 10%)
Due: Part A: September 29 (8:30a.m.), Part B: October 9 (10:00a.m.); in lecture.

Part A- Due September 29 (8:30a.m.)

---

1. (16 points) **Asymptotic notation properties.**
Let f and g be asymptotically positive functions. Prove or disprove each of the following conjectures. (In order to disprove a statement, it is sufficient to show a counter-example, as long as you prove that it violates the statement).

a. (4 points) $f \in O(g)$, where g is such that $g(n) = (f(n))^2$, for all $n \geq 0$.

b. (4 points) $f + g \in \Theta(f_{min})$, where $f_{min}(n) = \min\{f(n), g(n)\}$, for all $n \geq 0$.

c. (4 points) If $f \in O(g)$ then $g \in \Omega(f)$.

d. (4 points) If $f \in O(g)$ then $g \in O(f)$.

---

2. (30 points - 5/part) **Turing and RAM Machines and Analysis of Algorithms**
Define w^R to be the string which is the reverse of w. For example, if $w = 0001$ then $w^R = 1000$. Consider the language:
$L = \{w \in \{0,1\}^* : w \text{ and } w^R \text{ differ in every bit}\}$.

- Draw the state diagram for a 2-tape Turing machine M_2 that decides L.
- Analyse the running time for M_2; that is, state and prove an upper bound on the worst-case running time for M_2.
- Draw the state diagram for a single tape Turing machine M_1 that decides L.
- Analyse the running time for M_2; that is, state and prove an upper bound on the worst-case running time for M_1.
- Write a RAM program Π that decides L.
- Analyse the running time for Π, that is, state and prove an upper bound on the worst-case running time for Π.

Part B- Due October 9 (10:00a.m.)

3. (15 points) **Problem about P and NP**
Show that if $P = NP$ then every language $A \in P$, except for $A = \emptyset$ and $A = \Sigma^*$, is NP-complete.

4. (15 points) **Decision problems vs search problems.**
A *hamiltonian path* in a directed graph G is a simple directed path that visits every vertex exactly once. Define the language HAMPATH as follows:

$$\text{HAMPATH} = \{< G, u, v > \ : \ G = (V, E) \text{ is a directed graph, } u, v \in V \text{ and there exists a hamiltonian path from } u \text{ to } v \text{ in } G\}$$

Prove that if HAMPATH $\in P$, then there exists a polynomial time algorithm which actually finds the hamiltonian path (which is a sequence of vertices), if one exists.

Hint: To prove this, you need to design an algorithm which finds the hamiltonian path and show it runs in polynomial time. This algorithm can make calls to the algorithm that decides whether a hamiltonian path exists on a graph. The graphs used in these calls may not be the original graph.

5. (24 points) **NP-completeness proof**
Let DOUBLESAT $= \{< \phi >: \phi$ is a boolean formula with at least two satisfying assignments$\}$. Show that DOUBLESAT is NP-complete.

Hint: Use a reduction from SAT.

IMPORTANT:
Part of the marks will be for conciseness and clarity.
You **must read and comply** with the policy on plagiarism stated in the course web page.