Exam #1 - Data Structures and Algorithms
COMP 251-252 B - February 6, 2007

1. Simple Polygons

Three consecutive vertices *a*, *b*, *c* of a simple polygon form a *mouth* if *b* is a *concave* vertex and the closed triangle (*a*, *b*, *c*) does not contain any vertices of the polygon other than *a*, *b*, *c*. Prove or disprove that every non-convex simple polygon of *n* vertices, has at least one mouth.

2. Collapsing Compass

In proposition II of *Book I* of *The Elements*, Euclid proposed an algorithm (using the *straight-edge-and-collapsing-compass* computer) to *transfer* (or displace) a given distance [*a*,*b*] to a new location so that one end point, say *a*, lies on a pre-specified point *c* in the plane.

(a) Given a configuration consisting of a line segment [*b*,*c*] and a point *a*, design an algorithm that uses only the *collapsing-compass* (no straight edge is allowed) to construct a point *X* such that the line segment determined by the points *X* and *a* is the reflection (mirror image) of segment [*b*,*c*] across the line that perpendicularly bisects the points *a* and *b*. The *Lemoine simplicity* of such an algorithm is the number of circles drawn. What is the Lemoine simplicity of your algorithm? The lower the Lemoine simplicity of your algorithm, the higher the mark.

(b) Give a proof of correctness of your algorithm.

3. Algorithms for Turing Machines

Let $T(k)$ be an k-state Turing machine. Design an algorithm for $T(k)$ that reads a tape with $3n+1$ consecutive marks on it (where n is any arbitrarily large integer not smaller than 1) as input, and erases *all but* the $(n+1)$-*st* mark. Use the symbols *R*, *L*, *X* and - for: move right, move left, make a mark and make a blank, respectively. Assume the machine starts in state 1 and the reader is located on the leftmost mark on the tape. The machine should go to state 0 (the *stop* state) when finished. Let your instruction units (lines of code) be *4-tuples* of the form (*a*, *A*, *B*, *b*) where *a* denotes the present state, *A* takes on the symbols *X* or -, *B* takes on the symbols *R* or *L* or *X* or -, and *b* denotes the next state entered. The states *a* and *b* are integers. Example: if the machine is in state 5 and the reader is on a mark then move to the right one unit and go to state 7. The lower the value of *k* the higher the mark.

This question is to be answered only by COMP 252 students

--

1. Big "O" notation

Prove or disprove that if $f(n) = O(s(n))$ and $g(n) = O(r(n))$, then:

(a) $f(n) + g(n) = O(s(n) + r(n))$,

(b) $f(n) - g(n) = O(s(n) - r(n))$.