NAME:__________________________Student Number:___________________

CSE4/529 **MidTerm II** Fall, 2014

This exam is *closed book/notes/neighbors/etc.* Answer all questions on these exam pages. No code or pseudo-code is necessary – just a precise and concise explanation and justification. Give the best answer possible. *Unsupported work will receive no credit.*

Q1 of 3 (10 pts) Given a set of *n* **rainbow colors**, arbitrarily distributed, give an efficient algorithm to sort this set of *n* input values. Note that each of the *n* values is one of the seven rainbow colors: Red, Orange, Yellow, Green, Blue, Indigo, and Violet. Assume an ordering of the colors such that Red<Orange<Yellow<Green<Blue<Indigo<Violet.

Your algorithm *must* be based almost exclusively on use of the parallel prefix operation.

Discuss the i) asymptotic running time and ii) cost of your solution on each of the following architectures.

a) RAM
b) Mesh of size *n*
c) Mesh-of-Trees of base size *n*

NAME:____

Q2 of 3 (10 pts) Input: A set of n labeled line segments situated along the x-axis. Each line segment is initially represented by two records, one describing its left endpoint, as (x-value, label, **L**), and one describing its right endpoint, as (x-value, label, **R**). Assume that the $2n$ points are initially given ordered by x-value and that no two points have the same x-value.

Output: The number of breaks (*i.e.*, a single integer) between overlapping sequences of line segments in the range of the left endpoint of the first line segment to the right endpoint of the last line segment.

Discuss the i) asymptotic running time and ii) cost of your algorithm on the following architectures.

a) RAM
b) CREW PRAM

NAME:____

Q3 of 3 (10 pts) Input: A tree of base size *n* with a 1 or a 0 initially stored in every base processor.

Procedure: Give a *top-down divide-and-conquer solution* that runs in asymptotically optimal time on a tree of base size *n* to determine, for every processor in the tree, the number of 1's in the set of base processors at the base of its sub-tree.

Discuss the asymptotic running time of your algorithm.