# CSE 3302/5307 Lab Assignment 3

Due December 1, 2015

**Goal:**

Understanding of list manipulation in Scheme.

**Requirements:**

1. Write the following Scheme functions to implement *bags* of positive integers. A bag generalizes the notion of a set by having a positive cardinality (number of occurences) for each value in the set. Your bags will be ordered lists of pairs where the first element of a pair is the value and the second element is the cardinality. The ordered list for a bag is ordered ascending based on the values (first elements) of the pairs.

   a. `list2bag` - takes an unordered list of numbers (with duplicates) and produces a bag
   b. `bag?` - predicate to verify bag properties (elements are pairs, ordering, and positive cardinalities)
   c. `union` - takes the union of two bags by adding cardinalities. (A value that is not in a bag may be interpreted as having a cardinality of 0)
   d. `intersect` - takes the intersection of two bags by taking the minimum of cardinalities
   e. `diff` - analogous to set difference, but for two bags
   f. `symdiff` - analogous to the symmetric difference of sets, but for two bags
   g. `member? x mincard bag` - predicate to check that `bag` has a cardinality of at least `mincard` for `x`
   h. `countBag` - returns the sum of the cardinalities in a bag

2. Submit your Racket source file on Blackboard by 3:15 p.m. on Tuesday, December 1.

**Getting Started:**

1. Don't be concerned about efficiency.

2. Use of a few helper functions and `let` to avoid duplicate subexpressions can greatly simplify code.

3. This assignment may be done easily without `set!`.