UNBC

Laboratory Assignment

Computer Science 200—Fall 2006

Lists and Merge Sorting

Purpose:

The purposes of this assignment are: to learn how to use the list surgery functions of the STL `list` class, to learn how to implement merge sort on lists, and to understand problems related to manipulating list iterators.

Due Date:

This assignment is due *Friday, 1 December, 2006* at the beginning of class.

Assignment:

1. Implement merge sort for STL lists. You may use all of the STL supplied functions for lists, *except* `list<...>::merge` and `list<...>::sort`.
   (a) Use list surgery (`splice`) to move all of the elements, rather than using functions that would actually cause the elements to be copied.
   (b) Make your algorithms templated on the kind of data that it can receive, and on the order that it uses.
   (c) Use your random permutation construction algorithm from the previous assignment (or `stl::random_shuffle`) to create random `vector`s to create test data. Then use `stl::copy` to move your data.

   BONUS Implement a random shuffle algorithm that works directly on `list`s, and that runs in $o(n^2)$ time.
2. Test your sorting algorithms for *correctness*. To do this, write an automated test function that verifies that a list is sorted.
3. Test your sorting algorithms for *stability*. *Think about this early.* Again, write an automated test function.
4. Time your sorting procedures for various different sized data collections.
5. Compare the running time of your sorting algorithm with the running time of the STL `list<...>::sort` algorithm.
6. Graph your data to verify the Θ behaviour.