

CITS3001 Algorithms, Agents and Artificial Intelligence

Labsheet 1: sorting algorithms

1. Implement *insert sort* in the language of your choice. It's fine to use online sources for help, but make sure you give full credit for other people's work.
2. Build a testing framework for your program. The framework should generate (lots of) lists of random numbers of various lengths, then for each it should call insert sort and
 - a. check that the result list has the same elements as the original;
 - b. check that the result list is sorted in ascending order;
 - c. record the time taken for sorting;
 - d. tabulate the times against the lengths of the lists.
3. Implement an $O(n\log(n))$ sorting algorithm of your choice, e.g. *merge sort*, *quick sort*, or *heap sort*.
4. Use your testing framework to compare the performance of your $O(n\log(n))$ algorithm with *insert sort*.
5. Research *radix sort*, and write and test your own implementation.
6. Design and create some lists where radix sort will out-perform your $O(n\log(n))$ algorithm.