

CSI 3120 Assignment 1 Question 4

Student Name: Matt Langlois

Student Number: 7731813

Student Email: mlang025@uottawa.ca

Problem 1:

In question 1 we were asked to implement heapsort in Scheme. To implement heapsort you need to be able to set specific elements in the list when swapping element positions. Scheme doesn't offer an easy solution to just find the index, and set it. Instead we need to constantly recurse over the list making the readability of the function difficult. In fact, any operation that requires looking further than the first few elements into the list require recursive algorithms. Heapsort also requires functions to be performed on the list sequentially. Now while this can be done in scheme using the "begin" keyword it ends up creating nasty looking code.

If the problem were to be implemented in Java we would be able to make use of the pre-existing data structures to easily lookup indexes and set elements at a specific index without having to constantly recurse over the list just to find what we are looking for. Instead functions can be performed in a specific order sequentially using the clean syntax that Java provides.

Problem 2:

In question 2 we were asked to implement a one-bit full adder in prolog. Now while I didn't manage to complete this question in prolog I can discuss the pros/cons. Prolog was built as a logic language "Programming in logic" and low level circuits such as full adders are made from gates which just implement logic. Therefore, prolog would probably be the optimal choice to implement a full adder.

If a full adder were to be implemented in java it would require that all components of the full adder are created as objects. This means each type of gate, terminal and wire would need to be an object all linked together by a parent object, probably known as the circuit. This would create a lot of overhead in the design of the program and decrease the readability heavily. It makes a relatively simple program much more complex.