**LISP Sorting Report**

317203 Cairong Yang

**Task:**

In this laboratory 5 kinds of sorting algorithm will be implemented by LISP including quick sort, bubble sort, insert sort, selection sort and merge sort.

The environment used in this lab is GNU CLISP 2.49.92.

1.Recursive Function

**1.Quick sort:**

The algorithm is dividing an unsorted collection into two parts, all these items less than a particular item (i.e., Pivot). Pick out the first element and divide the list into two lists, one containing elements that should come before the first element, and one containing elements that should come after it. And then recursively sort the two shorter lists. Finally put everything back together in order. The recursion terminates when a list is empty.

* 1. **Recursive Function Code & Test Result**

Text

Description automatically generated

**2. Bubble sort**

Take the first element and compare it with the second one. If the first one is bigger then swap their position. If not, proceed and compare first one with the next. If first is bigger then swap their position otherwise moving on to next. If no one smaller than first one the first one finds it’s right position. And then do the process recursively starting from second element, each time the starting position is higher than the one in previous step. In the end ending up arranging all the elements in order.

**2.1 Recursive Function Code & Test Result**

Text

Description automatically generated

**3. Insert sort**

We have an original list to be sorted and set up an empty list. Keep taking items off the original list and insert them into the second list, each time we do so making sure we put the item we insert into the position in the second list which keeps that list in order. When the first list is empty the second list holds all its original items in sorted order.

**3.1 Recursive Function Code & Test Result**

Text

Description automatically generated

2.Recursive Macros

* 1. **Recursive Macros Code**

**2.1.1 Bubble sort**

Abstract the last-index, myprogn and swap as the Macros and reconstruct the bubble sort.

Text

Description automatically generated

**2.1.2 Selection sort**

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1. The subarray which is already sorted.
2. Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element from the unsorted subarray is picked and moved to the sorted subarray.

Text

Description automatically generated

**2.2.3 Merge sort**

Merge Sort is a [Divide and Conquer](https://www.geeksforgeeks.org/divide-and-conquer-introduction/) algorithm. It divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves.

**Text

Description automatically generated**

* 1. **Test Result**

Text

Description automatically generated