Thomas Herold

Professor Ramnath

CSCI 330-54

18 February 2025

Homework 3

Question 2:

The answers to parts 1-5 that mention defining/implementing functions are shown in "mergesort.lisp"

1. Q: What if there are zero or one item in the unprocessed list?
   A: **If there are zero elements left, both partitions should be returned. If there is one element left, it should be placed in the left partition.**
2. Q: As per mergesort, what should we do to each half?
   A: m**ergesort should be called recursively.**
3. Q: What are the end conditions?
   A: **If the left partition is empty, return the right partition. If the right partition is empty, return the left partition. If both partitions contain elements, compare the first elements and recursively merge them.**
4. This mergesort function is defined in "mergesort.lisp"

Question 4:

a. Q: While the insertion sort is in progress, we track 2 lists: the sorted items and the unsorted items. What should these look like when the process starts and when the process ends?
   A: **At the start, the sorted list is empty, and the unsorted list contains all the elements of the original input. At the end, the sorted list contains all the elements of the input list in sorted order, and the unsorted list is empty.**

b. Q: In each pass we start with two lists. At the end of the pass, we would have moved one more item from the unsorted list to the sorted list (fixed typo from assignment). When is this process trivially accomplished? What will the recursive call look like?
   A: **This process is trivially accomplished when the unsorted list is empty, which means all the elements should be inserted into the sorted list in the correct order. The recursive call moves one element at a time from the unsorted list to the sorted list.**
c. Q: To represent this insertion process, we need to track the items that have been examined, the item to be inserted and the items yet to be examined. There are two ways in which the process can terminate, what are they?
   A: **The process can terminate when unsorted is empty or when insterting into an empty sorted list.**