CSE310 Data Structure and Algorithm Project1-Milestone

Xiao Liu ASUID: 1211522751

9/15/2019

1 Encountered Problems

One problem I encountered while implementing the encoding based on the insertion sort is- how to take not only the 1^{st} character of the string into consideration but also the 2^{st} or more followed characters into the sorting algorithm. Since we are not allowed to use any other libraries, and we haven't cover the recursive function yet through the lecture, I used $while - loop$ with *goto* command so that when doing the insertion sorting, when I encountered two same 1^{st} characters in the string, I can compare the 2^{st} one and so on, until the $shift$ happens. Once the $shift$ happened, I use *goto* to enforce the algorithm to run based on the comparison of the 1^{st} character again. Therefore, the algorithms can return the sorting result correctly.

2 Bug Report

As for this Milestone submission, I didn't have any bugs for now. However, I would like to report some bugs during the production of this code.

(a) for cyclic shifts, one bugs report is core dump/segmentation fault.

(b) Access non allocated location of memory occurred when reading or extracting context from a location but contains nothing.

3 Interactions

I asked one question to TA- Lawrence on an index problem. ”How can I keep tracking the index so that I can just see where the previous index[0] - the original line - goes after sorting?”. I got the answer - using $std :: string :: compare()$ [1] to determine if a string is the same to the desired string which I am looking for.

Note that there is no other interaction between me and others when I produce my code.

References

[1] string::compare() in c++. https://www.geeksforgeeks.org/stdstringcompare-in-c/. Accessed: 2019-09-15.