UCR - CS 010C

Assignment 2

Deliverables:

Create a single PDF file that contains your answers to the questions. Then create a zip file that contains this PDF file along with all your code source files. Submit this zip file in iLearn.

Deadline: 10/27/2020 11:59 pm.

Exercise 1

If the elements of a list are sorted, is an array-based or a linked-list-based implementation of the list more efficient for binary search? Explain. Put your answer in the PDF file.

Exercise 2

Write a C++ class that implement two stacks using a single C++ array. That is, it should have functions pop_first(), pop_second(), push_first(…), push_second(…), size_first(), size_second(), …. When out of space, double the size of the array (similarly to what vector is doing).

Notes:

- Complete all the functions *in exercise_2.cpp*, then submit this cpp file.
- pop_first() and pop_second() should throw *std::out_of_range* exception when stack is empty.

Exercise 3

a. Implement functions for insertion sort, quicksort, heapsort and merge sort that input an array of integers and sort it in-place.
b. Write a program that generates random integer arrays (hint: use seed appropriately to avoid generating same sequences) of lengths 10, 100, 1000, 10,000, 100,000, 1000,000, and then sorts each using each of the sorting functions from (a), and measures the time in nanoseconds. The program will repeat this process 30 times and will compute the average execution time for each (arraysize,sorting-function) pair, over these 30 iterations. Finally, the program will output all these numbers in a readable format, e.g., as a table, in the PDF file.
c. Are your computed numbers reasonable given your knowledge of the asymptotic complexity of each sorting algorithm? Explain. Put your answer in the PDF file.

Note:

- Complete all the functions for a and b *in exercise_3.cpp*, then submit this cpp file.

Throughout the exercises, make any assumptions necessary.

CS 10C – HW 2

1. An array-based implementation of the list is more efficient for binary search because if it was a linked list, all the data would first have to be sorted. Even if it was sorted, binary search works by searching halves of data at a time, cutting it out when the item isn't found. This works better in an array, because the array is sorted in order, allowing for chunks of list items to be selected, searched, and cut out if the item isn't found. A linear search would work better for a linked list.

2. Exercise 2

3. The computer numbers are given my knowledge of asymptotic complexity of each sorting algorithm.