CptS 122 – Data Structures

Lab 11: Big-O and Merge Sort and More Practice with BSTs and Templates in C++

Assigned: Week of March 31, 2025
Due: At the end of the lab session

I. Learner Objectives:

At the conclusion of this programming assignment, participants should be able to:

- Implement the merge sort algorithm in C++
- Analyze the best-case, average-case, and worst-case scenarios for runtime of merge sort

II. Prerequisites:

Before starting this programming assignment, participants should be able to:

- Define Order of Magnitude (Big-O)
- Analyze algorithms and determine the best-case, average-case, and worst-case scenarios using Big-O

III. Overview & Requirements:

Labs are held in a "closed" environment such that you may ask your TA questions. Please use your TAs knowledge to your advantage. You are required to move at the pace set forth by your TA. Have a great time! Labs are a vital part to your education in CptS 122 so work diligently.

Tasks:

Recall (again ☺) with Big-O we want..

- **to determine central unit of work by considering the operations applied in the algorithm**
- **to express unit of work as function of size of input data: How quickly does amount of work grow as size of input grows?**
- **classify algorithms according to how their running *time* and/or *space* requirements grow as input size grows**
- **the analysis of an algorithm is extremely useful when comparing algorithms that solve the same problem!**

0. If you were not able to address the questions about Big-O last lab, please revisit the following questions. Provide a Big-O time and space analysis for each!

a. Is it more efficient to delete the last node in an array or linked implementation of a list?
b. Is it more efficient to delete the first node in an array or linked implementation of a list?
c. Is it more efficient to delete a node, in general, in an array or linked implementation of a list?
d. Is it more efficient to insert a node at the end in an array or linked implementation of a list?
e. Is it more efficient to insert a node at the front in an array or linked implementation of a list?
f. Is it more efficient to insert a node, in general, in an array or linked implementation of a list?
g. Is it more efficient to access a node, in general, in an array or linked implementation of a list?

1. Implement a C++ solution to merge sort. First, view the video at: https://www.youtube.com/watch?v=IR7DFDej6l4. Second, discuss in English an algorithm to describe the steps shown in the video. Third, implement a function template for your solution in C++. Once you've completed your solution analyze the best-case, average-case, and worst-case scenarios for merge sort.

2. Write an appropriate C++ member function template deleteNode() for a class template BST. The function must be able delete any node in the tree based on a generic, but comparable search value and then reconnect the links to Nodes correctly.

IV. Submitting Labs:

- You are not required to submit your lab solutions, unless you are unable to attend them synchronously. You should keep them in a folder that you may continue to access throughout the semester.

V. Grading Guidelines:

- This lab is worth 10 points. Your lab grade is assigned based on completeness and effort. To receive full credit for the lab you must show up on time, work in a team, complete 2/3 of the problems, and continue to work on the problems until the TA has dismissed you.