Texas Tech University Whitacre College of Engineering

CS 2413/5401 – Data Structures
Spring 2022
Lab Assignment 7

Acknowledge your collaborators or source of solutions, if any. **Online submission is required.**

While designing your programs or answering items, you are free to come up with your own assumptions based upon concepts and material learned in the course, if every potential specification is not given to you. Just be reasonable and document your assumptions. Such assumptions should not conflict with concepts and material learned in the course.

Your compliance with the "PROGRAMMING STYLE GUIDELINE" for CS 2413/5401 will affect your actual grade. All assignments will be checked for academic misconduct (cheating, plagiarism, collusion, falsifying academic records, misrepresenting facts, violations of published professional ethics/standards, and any act or attempted act designed to give unfair academic advantage to oneself or another student) defined by "OP 34.12: Grading Procedures, Including Academic Integrity" of TTU. Special software will be used to uncover such attempts.

A subset of answers submitted in this lab may be graded.

Objective: Practice heaps

Tasks:

1. You may work on this assignment by yourself, or you may work with one other student in this course as a team to complete this lab assignment.
    a. Teams larger than 2 people will incur a 25% penalty off the total lab points per extra person.
    b. It is expected that each team member contributes equitably and participates in coding and design ideas.
    c. If a team member is dissatisfied with the performance of the other team member, you are allowed to dissolve the team and continue individually.
        i. Whatever code each team member has contributed may be taken with that team member.
        ii. Try not to let such a decision wait for the day the assignment is due as no extensions will be given if a team is dissolved.
2. Write a C program, problem1.c, to merge n sorted files of integers named from 1.txt to n.txt using a binary min heap of size n. Each number in each file is on a separate line.
    a. Example
        i. Sorted files (shown on one line for brevity)
            1. 1.txt – 1 19 25
            2. 2.txt – 3 21 25 101
        ii. Binary Heap Array Trace (each node stores the integer input from the file and the file number)
            1. (1,1) (3,2) – output 1, read in 19
            2. (3,2) (19,1) – output 3, read in 21
            3. (19,1) (21,2) – output 19, read in 25
            4. (21,2) (25,1) – output 21, read in 25
            5. (25,2) (25,1) – output 25, read in 101
            6. (25,1) (101,2) – output 25, end of file 1
            7. (101,2) – output 101, end of file 2
        iii. Merged File (shown on one line for brevity)
            1. merged.txt – 1 3 19 21 25 25 101
    b. Use a binary min heap array to store one number from each file.

i. The binary min heap operations, such as insert, delete, and any others, should be written as separate functions.

c. Hints:
   i. bundle the heap data, the integer and its file number, as a struct to make heap data movements easier
   ii. use sprintf to make a filename, such as "1.txt" from the integer 1
   iii. use a FILE* array of size n to open the n files
   iv. use an open file flag array to track files that could not be opened or that have reached their end of file
   v. do not read from or write to the files in the binary min heap operations – let the binary min heap operations work only with the heap
   vi. insert and delete algorithms are in the chapter 12 slides for binary max heaps
   vii. use a define macro constant at the top of the program to set n (the number of files)

d. The main function should be a driver to call other functions to perform the required tasks.

e. No global variables should be used but define macro constants and typedef's may be used.

f. Report: In the comments at the end of the program, give the following information:
   i. Team Member Names
      1. For each team member, detail the work on the program concerning specific work, test cases, and code and functions designed, implemented, and modified, such as
         a. Name
            i. void insert (bin_t *, node_t); - designed/implemented/modified
            ii. created 1.txt, …, n.txt test cases
            iii. …
      2. A grade penalty of up to 25% of the total assignment points, which will be in addition to any other penalties, may be considered if inequitable contributions are made, not enough detail is present to be convincing, and/or all team members have the same list; i.e., all team members allegedly did the same thing (if both team members do the same thing – one team member is not needed).
   ii. Test Cases and Status
      1. example 1.txt, 2.txt, merged.txt built – passed/failed
      2. 3<=n<=12 files of sorted integers, merged.txt built – passed/failed
      3. one of the m.txt files of sorted integers is empty, merged.txt built – passed/failed
      4. one of the m.txt files of sorted integers is missing, merged.txt built – passed/failed
      5. …
   iii. Binary Heap Analysis
      1. Big O of inserting all m integers in files 1.txt to n.txt into a binary min heap of size n
      2. Big O of the storage requirements for a binary min heap of size n
      3. How could this program be extended to perform an external file merge sort on one file of m unordered integers?

Learning Outcomes:

- Understand C program concepts, such as structs and pointers
- Understand how to implement binary min heaps

Grading: 50 points

- Standard Deductions - 14 points
- Problem 1 – 36 points (problem1.c, 1.txt, 2.txt, …, n.txt)
  - Binary Heap – 12 points

- Report – 12 points
- Test Cases – 12 points

Due Date:

3/11/2022, 11:59pm (submitted on Blackboard by ONE team member)