CS3353: Data Structures and Algorithm Analysis I
Spring 2024

Homework #5

- Full name only: ______________________________________
- Release date: 4:00 PM, April 3, 2024 (Wednesday)
- Due date: **2:30 PM, April 17, 2024 (Wednesday)**
- It should be done INDIVIDUALLY; Show ALL your work; Submit your source code and results through Canvas.
- Please refer to the course syllabus (Course Requirements and Grading Policy - Assignment, page 3) for the policy of submission, late submission, missing submission, and wrong submission.
- Total: 20 pts
- **Grading Policy for Mandatary Question**
    - Compilation error / run-time error: -5 pts
        - Deduct 5 pts first. Then TA needs to evaluate the program and apply the following grading policy.
    - No homework submission: 0 pts
    - Creating binary tree: 4 pts
        - If the search results are wrong, probably the tree is built wrong.
    - Breadth-First Traversal: 3 pts
        - Correct output: 3 pts
        - Wrong output:
            - Partial points (e.g., 1 pt or 2 pts) are deducted depending on the degrees of wrongness.
    - Depth-First Traversal – preorder: 3 pts
        - Correct output: receive 3 pts
        - Wrong output: receive 1 pt
    - Depth-First Traversal – Inorder: 3 pts
        - Correct output: receive 3 pts
        - Wrong output: receive 1 pt
    - Depth-First Traversal – postorder: 3 pts
        - Correct output: receive 3 pts
        - Wrong output: receive 1 pt
    - Program menu as required: 2 pts
    - Self-testing results: 2 pts
        - No testing conducted / No WORD document: -2
    - Use other programming languages other than Java and C++: Deduct 5 pts first. Then TA needs to evaluate the program and apply the above grading policy.
    - Students will not receive 0 pts if students spent time and effort on program and make the submission.
        - If the program has issues/problems, TA needs to evaluate student's program and gives partial points depending on the quality/completion of program.
    - The instructor will decide the grade policy of any scenario which is not covered by the above list. Meanwhile, please kindly contact the instructor if you have any questions regarding the grading policy.

- **Grading Policy for Extra/Bonus Question**
    - Compilation error / run-time error: -2 pts
        - Deduct 2 pts first. Then TA needs to evaluate the program and apply the following grading policy.
    - No bonus submission: 0 pts
    - Breadth-First Traversal: 4 pts
        - Correct output: receive 4 pts

    - Wrong output: receive 1.5 pts
  - Self-testing results: 1 pt
    - No testing conducted / No WORD document: -1
  - Use other programming languages other than Java and C++: Deduct 3 pts first. Then TA needs to evaluate the program and apply the above grading policy.
  - Students will not receive 0 pts if students spent time and effort on program and make the submission.
    - If the program has issues/problems, TA needs to evaluate student's program and gives partial points depending on the quality/completion of program.
  - The instructor will decide the grade policy of any scenario which is not covered by the above list. Meanwhile, please kindly contact the instructor if you have any questions regarding the grading policy.

1. Write a program to build a binary tree from a sequence of data. Once the tree is constructed, conduct search and tree traversal functions including breadth-first traversal and depth-first traversal (preorder, inorder, and postorder). Search and tree traversal functions are applied to the most recently constructed tree. Here is a set of requirements to follow:

- Type the homework number and your full name at the top in your source code.

```
/* Homework #5, James Bond */
```

- Your program should be menu-driven and execute the chosen command. If you type 6, then exit the program.

```
                           M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                           Choose?
```

- Display a message, in case when searching a node that does not exist in the tree.
- Show ALL your work. For example,

```
                          M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  0  15  4  1  20  25  16


                          M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  1  35

There is no such node in the tree!
```

```
                          M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  2

15  4  20  1  16  25

                          M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  3

15  4  1  20  16  25


                          M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  4

1  4  15  16  20  25


Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Exit Program (6)

                          Choose?  5

1  4  16  25  20  15
                  .
                  .
                  .
```

- Please refer lecture slides and source code/pseudocode for creating binary tree from starch, Breadth-First Traversal, and Depth-First Traversal.

- Submit your source code and self-testing results (e.g., readable and clear screenshots) through Canvas before the due date, **2:30 PM, April 17, 2024 (Wednesday)**. The TA will build and run your source code and test with random input.
  - **Source code (one file only)** – The file name should be "your name + homework number", e.g., james_bond_5.cpp or james_bond_5.java.
  - **Self-testing Results** (e.g., readable and clear screenshots) **in WORD** document

2. [Extra/Bonus Credit] If you can implement a delete operation, extra 5 points will be provided. The program should deal with three deletion cases: no children, one child, and two children. Your menu should be shown below,

```
                        M E N U

Create (0), Search (1), Breadth-First Traversal (2)
Depth-First Traversal: preorder (3), inorder (4), postorder (5)
Delete (6), Exit Program (7)

                       Choose?
               .
               .
               .
```

- Submit your source code and self-testing results (e.g., readable and clear screenshots) of bonus program through Canvas before the due date, **2:30 PM, April 17, 2024 (Wednesday)**. The TA will build and run your source code and test with random input.
  - **Source code (one file only)** – The file name should be "your name + homework number", e.g., james_bond_5_bous.cpp or james_bond_5_bous.java.
  - **Self-testing Results** (e.g., readable and clear screenshots) **in WORD** document