CS2400 Project #2 (200 Points)
Spring - 2019
Instructor: Tannaz R.Damavandi
Due Date: Wed - 04/17/2019 at 11:59 pm.

Purpose:

1. Understand the structure and application of a Binary Search Tree (BST).
2. Implement Binary Search Tree using node representation (linked implementation).
3. Understand the structure of a binary heap.
4. Analyze the complexities of algorithms through experiment.

Task #1: Binary Search Trees (100 Pts)

Write a program in java which reads a sequence of integer values from input - values should be separated by space- and does the followings:

- Build a **binary search tree** using these values in the order they are entered.
- Print it in 3 traversal methods: pre-order, in-order, and post-order.
- Allow the user to Add or Remove a value. Once a new tree is generated, print it in in-order traversal.

NOTES:

- Use ***recursion*** to implement add and remove methods, otherwise you will lose points.
- Duplicates are NOT allowed in this BST. At the very first step after obtaining initial values check for duplicated values. In case of any duplication asks user to enter values again.
- Your program should use an interactive interface with the format shown in the following demo example (the user inputs are underlined):

```
% java Project2
Please enter values:
26 34 19 12 40 51 29 44 77 60 84 11 9 9 41 36 22 16 15
Duplication values are not allowed! Please enter values:
26 34 19 12 40 51 29 44 77 60 84 11 9 41 36 22 16 15
Pre-order: X X X ... X
In-order:  X X X ... X
Post-order:  X X X ... X
Main Menu
A : Add a value
R : Remove a value
E : Exit the program
What command would like to run? A
Please enter a value to add: 27
In-order:  X X X ... X
What command would like to run? A
Please enter a value to add: 11
11 already exists! Duplicated values are not allowed.
What command would like to run? R
Please enter a value to remove:12
In-order:  X X X ... X
What command would like to run? R
Please enter a value to remove: 10
10 doesn't exist!
What command would like to run? E
Exit!
%
```

You should test your program with the above data set as well as your own data sets. For the output submission, please use exactly the same data set shown in the demo example.

Task #2 Heap (100 Pts)

Write a program to build a max-heap using an array.

Task #2 – Part#1

- Create a basic user interface and allow user to select one of the following two options (Note that your need to implement both options):
  - (1) Test your program with 5 sets of 100 random positive integers in range 1 to 1000. (No duplicates are allowed).
  - (2) Test your program with some random positive integers entered by user (no duplicates are allowed)

- Implement both methods of building a max heap:
  - Using sequential insertions.
  - Using the optimal method.

  For both methods, you need to keep track of number of swaps (swapping parent and child) required to build a heap.

- For option (1), generate 5 sets of randomly generated integers in range 1 to 1000; compute, print and document (in your project report) the average number of swaps for both methods. Your program should output the average number of swaps for both methods (an average over 5 sets).

- For option (2), your program should ask user for some positive integers and show the number of swaps and level order tree traversal result.

Task #2- Part#2

- Record and report average number of swaps for both methods (over 5 sets) in case of option 1 and analyze heap implementation efficiency theoretically and experimentally. In case of option 2, enter some random number of your choice and output the number of swaps and level order traversal of the heap.
- Note: this report should be combined with the output of task#1 and submitted in pdf format.

What to Submit?

1- Java source code and .jar file **(Please comment your code properly)**
2- A detailed report in .pdf format as instructed inTask#2-Part#2 combined with output of task#1.
3- Readme.txt on how to run your code.
4- **Please zip all documents as yourname CS2400 Project2.zip and submit it on blackboard via provided link no later than due date/time.**

Discussion among students is encouraged, but I expect each student to hand in original work.