Design and Analysis of Computer Algorithms

Homework #4: Due 11:59PM, May 8th, 2022 (Sunday).

Problem #1 (10 points). Programming

Ice Cream Man

A young man likes ice cream very much. He plans to go to Baskin-Robbins to buy some ice cream. Since having a limited amount of money, he does not buy too expensive ice cream. Besides, he will not buy too cheap ice cream because it tastes not good. The store has n different kinds of ice cream and the price of i-th kind is p_i. He considers a kind of ice cream to be too expensive if its price is more than h and too cheap if its prices is less than l. The maximum amount of money he plans to spend is m.

What is the maximum number of kinds of ice cream he can buy?

Input and output:

Input is read from the text file **input.txt** consisting of:

- The first line contains $n,\ l,\ h,\ m$
- The second line includes a sequence $p_1, p_2, \ldots, p_n$ – the prices of n kinds of ice cream

Output is written to the text file **output.txt** consisting of ONLY ONE NUMBER which is the maximum kinds of ice cream he can buy.

Example:

input.txt	output.txt
3 1 100 100 50 100 50	2

Limitation:

- $1 \leq n \leq 100$
- $1 \leq l \leq h \leq 10^9$
- $1 \leq m \leq 10^9$
- Processing time < 1 second

Notice:

- Using C, C++, Java, or Python
- The name of the source code file is **icecream.xxx** (The extension **xxx** depends on the used programming language)

Problem #2 (50 points). Programming

AVL tree implementation

- Create a struct/class called **Student** with 3 attributes: *student_id*, *score*, and *age.* These attributes are non-negative integers and distinct among instances.
- Given a set of instances of struct/class Student, you are required to write a program that can construct an AVL tree based on one of the attributes (only one attribute is used as key to construct the key) and print the tree in **preorder**. No matter which attribute is selected as the key, only print *student_id* corresponding to the key. The program also need to be able to delete a node from the tree based on the key value.

Note:

- You can add more attributes to struct/class Student such as height, balance factor, parent, left child, and right child.
- You can create more structs/classes if necessary
- Basically, your programs will include following functions (each node is an instance of struct/class Student), you can add more if necessary

Function name	Inputs	Purpose
construct_avl()	a set of nodes	construct an AVL tree from a set of nodes
insert_node()	root node and node to be inserted	insert a node to the tree
delete_node()	root node and key value of the node to be deleted	delete a node from the tree
get_height()	node	get height of a node
balance_factor()	node	get the balance factor of a node
rotate_left()	node	perform left rotation
rotate_right()	node	perform right rotation
print_preorder()	root	print the tree in preorder traversal

Input and output:

- Input is read from **input.txt** file consisting of:
  - The first line contains a number denoting which attribute is selected as the key to construct the tree: 1 – student_id, 2 – score, 3 – age
  - The second line includes number of students n
  - The next n lines include 3 values each: student_id, score, and age
  - The last line contains a key value to be deleted
- Output is written to **output.txt** file which contains 2 lines
  - The first line contains n student id of the constructed AVL tree (preorder)

- The second line contains $(n-1)$ student id of the tree after deleting one key (preorder)

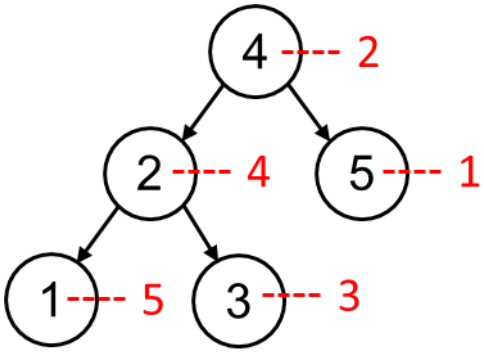
Example:

input.txt	output.txt
2 5 1 5 1 2 4 5 3 3 2 4 2 4 5 1 3 5	2 4 5 3 1 4 5 2 3

Example explanation:

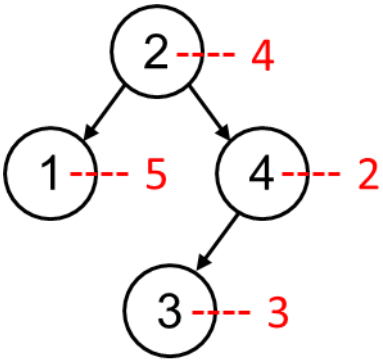
- The value in the first line is 2, which means you need to construct the tree based on *score* attribute
- There are 5 students, their scofres are: 5 4 3 2 1

The AVL tree constructed based on the scores and the corresponding *student_id* (red numbers) are:



The preorder traversal in the output is 2 4 5 3 1

- The value in the last line is 5, which mean the node with score 5 will be deleted, the AVL tree after deleting and its corresponding *student_id* are:



The preorder in the output is 4 5 2 3

Limitations:

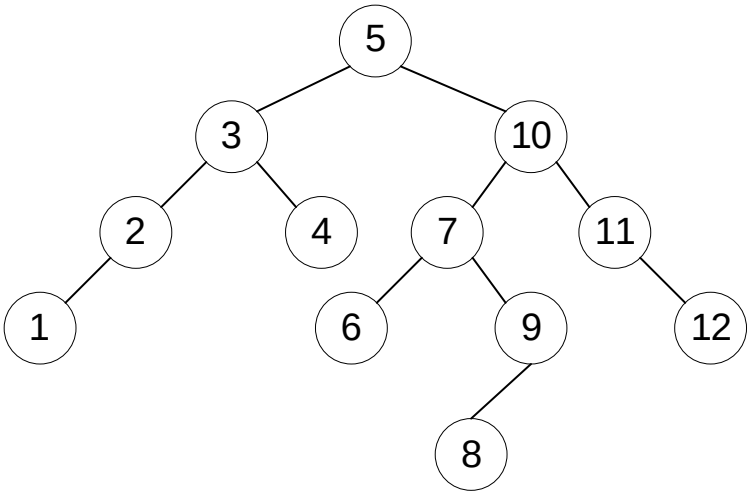
- 1 ≤ number of students ≤ 100

Notice:

- Using C, C++, Java, or Python
- The name of the source code file is **avl.xxx** (The extension **xxx** depends on the used programming language)
- You can modify an open source code to adapt with this problem. Submit both the original source code and your modified code in separated folders

Problem #3 (10 pts). Insert items with the following keys (in the given order) into an initially empty binary search tree: **24, 40, 30, 58, 48, 13, 11, 26**. Draw the tree after each insertion.

Problem #4 (10 pts): You are given an AVL trees as follows:



a) After eliminating element "5", redraw the binary search tree (*with balance factor* for each node) before any re-balancing procedures.
b) Now draw the rebalanced tree that results from (a) after rebalancing the tree. You do not need to label these trees with balance factors.

Problem #5 (10 pts). You are given six matrices with their sizes as follows:

$$\mathbf{A_1(5,4),\ A_2(4,6),\ A_3(6,2),\ A_4(2,7)}$$

Determine an order for multiplying all matrices ($A_1A_2A_3A_4$) that has the lowest cost (optimal way to do parenthesizations). Explain your answer clearly by creating two matrices m and s in slide # 16 and 18.

Problem #6 (10 pts). Prove or disprove the following statement: “In a binary search tree, predecessor and successor of a node which has two children does not have right child and left child, respectively.”

What you have to submit:

1) Your source programs and executable files. For problem #2, submit both original source code and your modified code
2) Your input and output files
3) Documentation file (**HW4.DOCX**)
   - Solution of the assigned problems
   - Write the explanation about your implementation

◆ Submit your compressed file named as HW4_ID_NAME.zip

(ex. HW4_2013711123_홍길동.zip) to iCampus

NOTICE:

✓ BOTH ORIGINAL AND COPY WILL GET –30 POINTS EACH INSTEAD OF 0S
✓ ANY SOURCE CODE WITH COMPILE OR RUNTIME ERROR WILL GIVE YOU 0 POINTS
✓ THERE WILL BE POINTS OFF FOR INAPPROPRIATE SUBMISSION STYLE
✓ ALL THE HOMEWORK MATERIALS (INCLUDING EMAIL CONTENTS AND DOCUMENTATION) SHOULD BE MADE IN **ENGLISH**

Good luck!