Indian Statistical Institute

Semester-I 2024–2025

M.Tech.(CS) - First Year

Lab Test 4 (06 December, 2024)

Subject: Computing Laboratory

Total: 60 marks **Duration: 4 hrs.**

INSTRUCTIONS

1. You may consult or use slides / programs provided to you as course material, or programs that you have written yourself as part of classwork / homework for this course, but please **do not** consult or use material from other Internet sources, your classmates, or anyone else.
2. Unless otherwise specified, all programs should take the required inputs from stdin, and print the desired outputs to stdout. Please make sure that your programs adhere strictly to the specified input and output format. **Your program may not pass the test cases provided, if your program violates the input and output requirements.**
3. Submissions from different students having significant match will **not be evaluated**.
4. To avoid mismatches between your output and the provided output, please store all floating point numbers in `double` type variables.

Part I

You are given a list of N records, each corresponding to a film. Each record consists of the following 7 fields: the film's title, the year it was released, its running time in minutes, genre tags assigned to the film, its average IMDB rating, the number of users who rated it, and a unique identifier corresponding to the film. The unique identifiers are 9 or 10 character strings, starting with `tt`, followed by distinct 7–8 digit integers (possibly with leading zeroes). The fields are separated by semi-colons (";"). You may assume that a semi-colon never occurs *within* the data in any field.

Q1. (5 marks)

Write a program to output the number of comparisons required to search for a film using the unique identifier if the records are stored in an unsorted array.

Q2. (10 marks)

Write a program to output the number of comparisons required to search for a film using the unique identifier if the records are stored in an ordinary binary search tree (BST).

Q3. (20 marks)

Write a program to output the number of comparisons required to search for a film using the unique identifier if the records are stored in an AVL tree.

The array stores the records in the same order as they occur in the input. The BST and AVL tree are initially empty, and are constructed by inserting the records one by one as they occur in the input.

Input format: N, the total number of records, followed by the records in the format described above, one per line, followed by m unique identifiers (search keys), one per line. The value of m will not be given to you, but you may assume that no single line contains more than 125 characters.

Output format: For each of the m search keys, your program should print 2 lines:

1. the first line should contain the data item matching the search key, or the string `No match found`;
2. the second line should be one of the following, depending on the data structure that you have used in your program:
   (a) the index of the matching item in the unsorted array (a single integer between 0 and $N - 1$), or N if the search key does not match any of the N data items;
   (b) the path from the root to the node matching the search key in the (plain) BST, followed by the depth of this node;
   (c) the path from the root to the node matching the search key in the balanced BST, followed by the depth of this node.

Notes:

- The matching items should be printed in the same format in which they are provided in the input.
- The path from the root to a node should be a single string consisting of the letters `L` and/or `R`, specifying which branch needs to be taken at each node, starting with the root.
- For parts (b) and (c), if the search key does not match any of the N data items, the path and depth should correspond to the node where your program determines that the search failed.

Sample input 0: (this is the second test case provided to you)

```
10
Planet of the Apes;2001;120;Action,Adventure,Sci-Fi;5.7;236260;tt0133152
The Change-Up;2011;112;Comedy,Fantasy;6.3;199157;tt1488555
The Bone Collector;1999;118;Crime,Drama,Mystery;6.7;190114;tt0145681
Alien:  Covenant;2017;122;Horror,Sci-Fi,Thriller;6.4;323011;tt2316204
The Book Thief;2013;131;Drama,War;7.5;143934;tt0816442
Casanova;2005;112;Adventure,Comedy,Drama;6.5;56857;tt0402894
Cape Fear;1991;128;Crime,Thriller;7.3;222578;tt0101540
Dressed to Kill;1980;104;Crime,Drama,Mystery;7.1;51314;tt0080661
Rushmore;1998;93;Comedy,Drama,Romance;7.6;200983;tt0128445
Texas Chainsaw;2013;92;Horror,Thriller;4.8;57444;tt1572315
tt0145681
tt1572315
tt0213847
tt0101540
```

Sample output 0 for Q1:

```
The Bone Collector;1999;118;Crime,Drama,Mystery;6.7;190114;tt0145681
2
Texas Chainsaw;2013;92;Horror,Thriller;4.8;57444;tt1572315
9
No match found
10
Cape Fear;1991;128;Crime,Thriller;7.3;222578;tt0101540
6
```

Sample output 0 for Q2:

```
The Bone Collector;1999;118;Crime,Drama,Mystery;6.7;190114;tt0145681
RL 2
Texas Chainsaw;2013;92;Horror,Thriller;4.8;57444;tt1572315
RRL 3
RLRLL 5
Cape Fear;1991;128;Crime,Thriller;7.3;222578;tt0101540
L 1
```

Explanation: Your tree should look like the following:

Index	Left	Right	Key
0	6	1	tt0133152 (Apes)
1	2	3	tt1488555 (Change-Up)
2	-1	4	tt0145681 (Bone)
3	9	-1	tt2316204 (Alien)
4	5	-1	tt0816442 (Book)
5	-1	-1	tt0402894 (Casanova)
6	7	8	tt0101540 (Cape Fear)
7	-1	-1	tt0080661 (Dressed)
8	-1	-1	tt0128445 (Rushmore)
9	-1	-1	tt1572315 (Texas)

Sample output 0 for Q3:

Q4. (15 marks)

Consider a binary tree with at least two leaves. The nodes of the tree store integers. Given two leaves u_1 and u_2, $\boldsymbol{PathSum}(u_1, u_2)$ is defined as the sum of the integers stored in the nodes on the path from u_1 to u_2 (both leaves are also included). Write a program to find the maximum value of $\boldsymbol{PathSum}(v_1, v_2)$ across all pairs (v_1, v_2) of leaves in the tree.

Input format: The input will consist of a positive integer N, followed by N lines. Each of these lines will correspond to a node of a binary tree, and will consist of 3 integers: the data stored in the node, the index of its left child, and the index of its right child of the node[1]. If a node does not have a left/right child, the corresponding index will have a value of -1. Valid indices have values from $\{0, 1, 2, \ldots, N-1\}$.

Output format: If ***PathSum*** is maximum for the pair (w_1, w_2) of leaves, you should print the indices of w_1 and w_2 in the tree, along with value of ***PathSum***(w_1, w_2).

Q5. (10 marks)

Write a program which, given the (fictitious) birth and death dates of famous Xollywood film stars, determines the largest number of stars who were alive simultaneously, and the period during which they were all alive. If a person is alive for a part of a day, include that day in the period. You may assume that all births occurred at 02:00 PM, and all deaths occurred at 05:30 AM. Thus, if one star's birth date coincides with another star's date of death, the two have **not** been alive simultaneously.

Input format: N lines of text (the value of N will not be given to you beforehand), each of which will contain a film star's name (consisting only of letters and spaces) and his/her birth and death dates in DD.MM.YYYY format. For this problem, assume that all months (including February) have 30 days.

Output format: The largest number of stars who were alive simultaneously (a positive integer), followed by the earliest and the last dates (also in DD.MM.YYYY format) during which they were all alive. If there are ties, i.e., the same maximum number of simultaneously alive stars occurred during multiple (disjoint) periods in history, print the earliest period.

Sample input 0: (this is the first test case provided to you)

```
Balraj Taylor 01.05.1913 13.04.1973
Suchitra di Caprio 06.04.1931 17.01.2014
MR Ganesan Rao 28.05.1923 24.12.1987
Amitabh Chatterjee 03.09.1926 24.07.1980
Uttam Law 10.02.1997 31.12.2091
```

Sample output 0: `4 06.04.1931 13.04.1973`

[1] You can read the tree using the `read_tree()` function discussed in class.