General Note:

- There is no design submission for this question.
- For implementation, you are permitted to use the code that you have submitted as an assignment in Eduserver.
- Verify the correctness of your program with **the two sample test cases that are uploaded in Eduserver.**

3. Consider two software applications (Apps) A_1 and A_2 that are running in a computer. Each App has a set of tasks to be performed, where each task is associated with a task ID t_id and CPU time $time$. Assume that each App has at least one task, and no two tasks in an App have the same $time$. Task details of each of the Apps are stored in a separate **Red-Black tree**.

   The *fairness factor* of a task T is defined as the maximum number of black node tasks on any simple path from T down to a leaf (not including T). Note that the *fairness factor* of an App is the *fairness factor* of its root task itself, and the *fairness factor* of a NIL node is defined as 0.

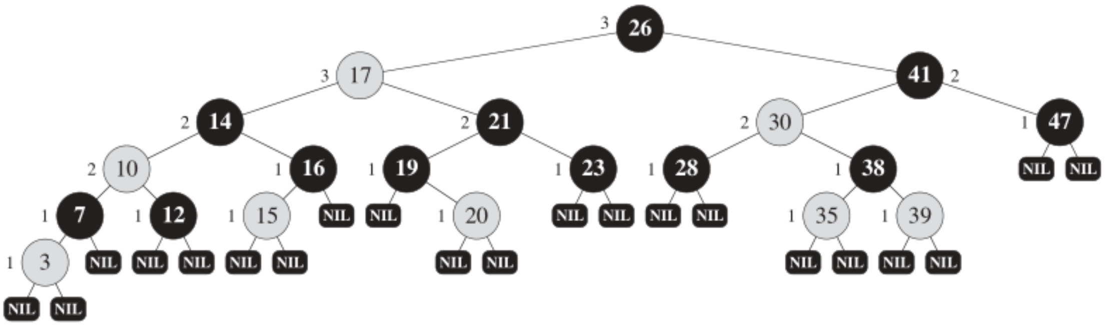

Figure shows the RB tree representing the App A2 with fairness factor 3 in the sample input 2 uploaded in the Eduserver. The value inside a node represents *time* of the corresponding task and the value next to the node represents the *fairness factor* of that task.

At any instant of time, CPU runs a single task T, and it is selected as follows:

- Select an App A from A_1 and A_2 that has the largest *fairness factor*, and the other App waits. If both A_1 and A_2 have the same *fairness factor*, select A = A_1.
- Select all the tasks in A with *fairness factor* equal to that of the waiting App.
- Out of them, the task with highest CPU time is assigned to the CPU.

Your program must include the following functions:

- $insert\_task(t\_id, time, A)$: Insert the task with task ID t_id and CPU time $time$ into the App A.
- $fairness\_factor(T)$: Return the *fairness factor* of the task T.
- $get\_application(A_1, A_2)$: Given two Apps A_1 and A_2, return the $app\_no \in \{1, 2\}$ of the App having the largest *fairness factor*, using the function $fairness\_factor()$. If both A_1 and A_2 have the same *fairness factor*, then return 1.

- $select\_task(A_1,\ A_2)$: Given the Apps A_1 and A_2, select the task T to be assigned to the CPU as described above, using the function $get\_application()$. Print t_id of T.
- $print\_application(A)$: Print the App A in the parenthesis format as: (t_id $time$ color (left-subtree) (right-subtree)). A null tree is represented as (NIL B).

Input/Output Format

The input consists of multiple lines. Each line starts with a character from $\{i, a, g, s, p, t\}$ followed by zero or more integers or characters. The integers, if given, are in the range $[1, 10^5]$ and the characters, if given, are from [A-Z, a-z].

- Character 'i' : Character 'i' will be followed by a string t_id of maximum length 100, an integer $time$, and an integer $app\_no \in \{1, 2\}$ of a task T, separated by a space.
  - If app_no = 1, insert T into the App A_1, using $insert\_task()$ function.
  - If app_no = 2, insert T into the App A_2, using $insert\_task()$ function.
- Character 'a' : Character 'a' will be followed by an integer $k \in \{1, 2\}$ separated by a space. Print the*fairness factor* of the the App with app_no = k, using $fairness\_factor()$ function.
- Character 'g' : Print the $app\_no \in \{1, 2\}$ of the App having the largest *fairness factor*, using $get\_application()$ function.
- Character 's' : Print the $task\_id$ of the task selected using the $select\_task()$ function.
- Character 'p' : Character 'p' will be followed by an integer $app\_no \in \{1, 2\}$ separated by a space.
  - If app_no = 1, print App A_1, using $print\_application()$ function.
  - If app_no = 2, print App A_2, using $print\_application()$ function.
- Character 't' : Terminate the program.

Sample Input and Output

Input 1

```
i T11 6 1
i T12 7 1
i T13 8 1
i T21 10 2
i T22 18 2
i T23 7  2
i T24 15 2
p 1
p 2
a 1
a 2
g
s
t
```

Output 1

```
( T12 7 B ( T11 6 R ( NIL B ) ( NIL B ) ) (T13 8 R ( NIL B ) ( NIL B
) ) )
( T21 10 B ( T23 7 B ( NIL B ) ( NIL B ) ) ( T22 18 B ( T24 15 R ( NIL
B ) ( NIL B ) ) ( NIL B ) ) )
1
2
2
T22
```