

# Assignment 3

---

(60 marks)

## Problem Statement:

The massive increase in number of police vehicles trying to track down the suspects of the bank robbery is making our system slow. From now onwards, every time a police vehicle leaves our intersection, they will automatically send a request to delete their own vehicle number from our set. So we would like to allow deletion as well as dynamic insertion of elements into the set we are maintaining.

Independently, we would like to convince the city administration to use our software to monitor all intersections in the city road network. They would want to test our program by verifying the list of elements in our set at various points during a run. Hence we would like to allow for an inorder and postorder traversal of our BST whenever requested in the input.

## Input Format:

- The first line of the input will give a list of distinct numberplates  $a_1, a_2, \dots, a_n$  with each  $a_i$  separated by space. This line ends with a `\n` character.
- Every following line starts with one of: `S`, `<`, `>`, `-`, `+`, `I`, `P`, and is followed by a number  $m$ .
- End of input is indicated by EOD character.

## Output:

No output against the first line of input. Simply build a BST out of those elements.

Let the input line read be: `op m`

- If `op`  $\in$  `{ S, <, > }`, see previous assignment documents for correct output.
- If `op` = `+`, then insert  $m$  into your BST if it does not already exist. No output.
- If `op` = `-`, then delete  $m$  if it exists in your BST. Output should be as follows:
  - Print the number of children of  $m$  followed by a space.
  - If the number of children is greater than 0, then print the value of the node that takes  $m$ 's place.
  - End the line with `\n`.

If  $m$  does not exist in your BST, then output "-1" followed by `\n`.

- If `op` = `I`, then  $m$  plays no role. Do the following:
  - Print all elements using an inorder traversal with `space` as delimiter between elements.
  - End with `\n`.
- If `op` = `P`, then  $m$  plays no role. Do the following:
  - Print all elements using a postorder traversal with `space` as delimiter between elements.
  - End with `\n`.

Read the next line and repeat the above. Your program should terminate only when input is EOF.

## Implementation Rules:

- You have to store  $a_1, a_2, \dots, a_n$  in a Binary Search Tree.
- When deleting a node with 2 children, replace it with the successor.
- Nodes deleted from your BST have to be deleted from memory.
- All operations have to be done on the BST.

## Remarks

- Write a separate splice function and reuse it every time you need to splice.
- You are encouraged to use your own code from your previous assignment submissions.