SCS1308: Foundations of Algorithm

Programming Question: Implement a Dictionary using a Binary Search Tree

1. **Objective:**
   Create a program to build a dictionary using a binary search tree (BST).

2. **Input:**

   - The program should read a list of words from a **file input** (not hardcoded).
   - Use the following words for the input file:
     Cat, Dog, Rat, Bat, Bag, Nut, Mug, Cup, Kid, Dig.

3. **Tasks to Perform:**
   Implement the following functionalities for the constructed binary search tree:

a. **FindHeight()**: Compute and return the height of the tree.
b. **FindSize()**: Compute and return the total number of nodes in the tree.
c. **FindMin()**: Find and return the node with the smallest value.
d. **FindMax()**: Find and return the node with the largest value.
e. **MirrorImage()**: Create the mirror image of the tree and print it using proper spacing to visualize the structure.

4. **Output:**

   - Print the results of the functions: FindHeight, FindSize, FindMin, and FindMax.
   - For MirrorImage, print the mirrored tree in a readable format.

5. **Requirements:**

   - Ensure the program dynamically builds the tree from the file input.
   - Use proper tree traversal methods (e.g., in-order, pre-order) to process and display the results where appropriate.
   - Avoid hardcoding the input list into the code.

Constraints:

1. Node names are single lower letters (a-z).
2. Graph may contain up to 100 node

Submission guidelines:

1. Submit your code (in c language) by **13th Feb 2025 by 6pm.**
2. Hand in your source code electronically (do not submit .o or executable code).
3. Each student uploads only ONE copy of the assignment
4. Make sure that this code compiles and runs correctly on linux. The makefile must give the executable code the name bst
5. Write a README file (text file, do not submit a .doc file) which contains

- Name, Index number and email address of the student
- Whether your code was tested on linux.
- How to execute your program.
- Briefly describe anything special about your submission that the instructor should take note of.