

Assignment_30-08-2024

1 Binary search tree

To practice gdb options and fix issues with gdb commands, let's first compile the given C++ code with the `-g` flag to include debugging symbols.

```
mladev@Moclananhh: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/1.bst$ g++ -g main.cpp -o binary_search_tree
mladev@Moclananhh: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/1.bst$ ll
total 40
drwxrwxrwx 1 mladev mladev 512 Aug 30 15:19 ./
drwxrwxrwx 1 mladev mladev 512 Aug 30 14:53 ../
-rwxrwxrwx 1 mladev mladev 33928 Aug 30 15:19 binary_search_tree*
-rwxrwxrwx 1 mladev mladev 1740 Aug 30 14:56 main.cpp*
mladev@Moclananhh: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/1.bst$ gdb ./binary_search_tree
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./binary_search_tree ...
(gdb)
```

Since the code seems to be incomplete, let's identify the issue. We notice that the insert function is missing the logic to handle the case when the key to be inserted is smaller than the node's key.

This is updated version:

```
Node *insert(Node *node, int key)
{
    // If the tree is empty, return a new node
    if (node == NULL)
        return new Node(key);
    // If the key is already present in the tree,
    // return the node
    if (node->key == key)
        return node;
    // Otherwise, recur down the tree/ If the key
    // to be inserted is greater than the node's key,
    // insert it in the right subtree
    if (node->key < key)
        node->right = insert(node->right, key);
    // If the key to be inserted is smaller than
    // the node's key, insert it in the left subtree
    else
        node->left = insert(node->left, key);
    // Return the (unchanged) node pointer
    return node;
}
```

```

Breakpoint 1, main () at main.cpp:69
69      Node *root = new Node(50);
(gdb) n
70      root = insert(root, 30);
(gdb) n
71      root = insert(root, 20);
(gdb) n
72      root = insert(root, 40);
(gdb) n
73      root = insert(root, 70);
(gdb) n
74      root = insert(root, 60);
(gdb) n
75      root = insert(root, 80);
(gdb) n
78      inorder(root);
(gdb) n
80      return 0;
(gdb) n
81  }
(gdb) n
__libc_start_call_main (main=main@entry=0x555555552cb <main()>, argc=
argc@entry=1, argv=argv@entry=0x7fffffffd38) at ../sysdeps/nptl/libc_
start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or direct
ory.
(gdb) n
20 30 40 50 60 70 80 [Inferior 1 (process 9806) exited normally]
(gdb)

```

2 Min max

Compile the program and go to debug mode to check code

```

mladev@Moclananhh:/mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max$ ll
total 1
drwxrwxrwx 1 mladev mladev 512 Aug 30 15:46 ./
drwxrwxrwx 1 mladev mladev 512 Aug 30 15:40 ../
-rwxrwxrwx 1 mladev mladev 660 Aug 30 15:45 main.cpp*
mladev@Moclananhh:/mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max$ g++ main.cpp -o main -g
^[[Amladev@Moclananhh:/mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max$ gdb main
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from main...
(gdb) b main
Breakpoint 1 at 0x128a: file main.cpp, line 31.
(gdb) r
Starting program: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at main.cpp:31
31      {
(gdb)

```

```

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from main ...
(gdb) b main
Breakpoint 1 at 0x128a: file main.cpp, line 31.
(gdb) r
Starting program: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at main.cpp:31
31 {
(gdb) n
32     int A[] = {4, 9, 6, 5, 2, 3};
(gdb) n
33     int N = 6;
(gdb) n
34     cout << "Minimum element is: " << setmini(A, N) << endl;
(gdb) n
Minimum element is: 2147483647
35     cout << "Max element is: " << setmaxi(A, N) << endl;
(gdb) n
Max element is: -2147483648
36 }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555527d <main()>, argc=argc@entry=1, argv=argv@entry=0x7ffffffdc08) at
t_call_main.h:74
74  ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)

```

Issues:

- In the setmini function, the code is missing the assignment of the minimum value to mini.
- In the setmaxi function, the loop condition is incorrect. It should be $i < N$ instead of $i > N$.

Fixed code:

```

int setmini(int A[], int N)
{
    int mini = INT_MAX;
    for (int i = 0; i < N; i++)
    {
        if (A[i] < mini)
        {
            mini = A[i]; // assign the minimum value to mini
        }
    }
    return mini;
}

int setmaxi(int A[], int N)
{
    int maxi = INT_MIN;
    for (int i = 0; i < N; i++) // fix the loop condition
    {
        if (A[i] > maxi)
        {
            maxi = A[i];
        }
    }
    return maxi;
}

```

```

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from main ...
(gdb) b main
Breakpoint 1 at 0x12be: file main.cpp, line 32.
(gdb) run
Starting program: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/2.min_max/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at main.cpp:32
32  {
(gdb) n
33      int A[] = {4, 9, 6, 5, 2, 3};
(gdb) n
34      int N = 6;
(gdb) n
35      cout << "Minimum element is: " << setmini(A, N) << endl;
(gdb) n
Minimum element is: 2
36      cout << "Max element is: " << setmaxi(A, N) << endl;
(gdb) n
Max element is: 9
37  }
(gdb) n
__libc_start_call_main (main=main@entry=0x5555555552b1 <main()>, argc=argc@entry=1, argv=argv@entry=0x7ffffffdc08) at
t_call_main.h:74
74  ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb)

```

3 tree

Run this code to check

```

aladev@Moclananh: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/3.tree$ g++ main.cpp -o main -g
main.cpp: In function 'void printLeafNodes(int, std::vector<std::vector<int>> &adj)':
main.cpp:64:32: error: lvalue required as left operand of assignment
   64 |         if (adj[i].size() == 1 && i = Root)
      |                                ^
aladev@Moclananh: /mnt/d/WSL2/Coding/2.CPP/2.Coding/1.AssignmentCode/1.Assignment_1/19.Debugging/3.tree$

```

Problem here:

1. Logical Error in printLeafNodes Function

- The condition `if (adj[i].size() == 1 && i = Root)` is incorrect. It should be `if (adj[i].size() == 1 && i != Root)` to correctly identify leaf nodes. Additionally, `=` is an assignment operator and should be `!=` for comparison.

2. Logical Error in printDegrees Function

- Similarly, the condition `if (i = Root)` should be `if (i == Root)`.

3. Array Initialization Issue in printChildren Function

- The initialization of the `vis` array using `{0}` is incorrect for dynamic arrays. Instead, we should use `fill_n` or `vector<int>` to handle this dynamically.

4. Root Node Condition in printParents Function

- The condition `if (parent == 0)` should check if the parent is the root, so `parent == -1` is a better choice for indicating the root node.

Code fixed:

```

#include <bits/stdc++.h>
using namespace std;

// Function to add an edge between vertices x and y
void addEdge(int x, int y, vector<vector<int>> &adj)
{
    adj[x].push_back(y);
    adj[y].push_back(x);
}

```



```

}
// Function to print the parent of each node
void printParents(int node, vector<vector<int>> &adj, int parent)
{
    // current node is Root, thus, has no parent
    if (parent == -1)
        cout << node << "->Root" << endl;
    else
        cout << node << "->" << parent << endl;
    // Using DFS
    for (auto cur : adj[node])
        if (cur != parent)
            printParents(cur, adj, node);
}
// Function to print the children of each node
void printChildren(int Root, vector<vector<int>> &adj)
{
    // Queue for the BFS
    queue<int> q;
    // pushing the root
    q.push(Root);
    // visit vector to keep track of nodes that have been visited
    vector<int> vis(adj.size(), 0);
    // BFS
    while (!q.empty())
    {
        int node = q.front();
        q.pop();
        vis[node] = 1;
        cout << node << "-> ";
        for (auto cur : adj[node])
        {
            if (vis[cur] == 0)
            {
                cout << cur << " ";
                q.push(cur);
            }
        }
        cout << endl;
    }
}
// Function to print the leaf nodes
void printLeafNodes(int Root, vector<vector<int>> &adj)
{
    // Leaf nodes have only one edge and are not the root
    for (int i = 1; i < adj.size(); i++)
        if (adj[i].size() == 1 && i != Root)
            cout << i << " ";
    cout << endl;
}
// Function to print the degrees of each node
void printDegrees(int Root, vector<vector<int>> &adj)
{
    for (int i = 1; i < adj.size(); i++)
    {
        cout << i << ": ";
        // Root has no parent, thus, its degree is equal to the edges it is connected to
        if (i == Root)
            cout << adj[i].size() << endl;
        else
            cout << adj[i].size() - 1 << endl;
    }
}

```

```

    }
}
// Driver code
int main()
{
    // Number of nodes
    int N = 7, Root = 1;
    // Adjacency list to store the tree
    vector<vector<int>> adj(N + 1, vector<int>());
    // Creating the tree
    addEdge(1, 2, adj);
    addEdge(1, 3, adj);
    addEdge(1, 4, adj);
    addEdge(2, 5, adj);
    addEdge(2, 6, adj);
    addEdge(4, 7, adj);
    // Printing the parents of each node
    cout << "The parents of each node are:" << endl;
    printParents(Root, adj, -1); // -1 indicates that the root has no parent
    // Printing the children of each node
    cout << "The children of each node are:" << endl;
    printChildren(Root, adj);
    // Printing the leaf nodes in the tree
    cout << "The leaf nodes of the tree are:" << endl;
    printLeafNodes(Root, adj);
    // Printing the degrees of each node
    cout << "The degrees of each node are:" << endl;
    printDegrees(Root, adj);
}

```

```

100      cout << "The parents of each node are:" << endl;
(gdb) n
The parents of each node are:
101      printParents(Root, adj, -1); // -1 indicates that the root has no parent
(gdb) n
1→Root
2→1
5→2
6→2
3→1
4→1
7→4
104      cout << "The children of each node are:" << endl;
(gdb) n
The children of each node are:
105      printChildren(Root, adj);
(gdb) n
1→ 2 3 4
2→ 5 6
3→
4→ 7
5→
6→
7→
108      cout << "The leaf nodes of the tree are:" << endl;
(gdb) n
The leaf nodes of the tree are:
109      printLeafNodes(Root, adj);
(gdb) n
3 5 6 7

```

```
112     cout << "The degrees of each node are:" << endl;
(gdb) n
The degrees of each node are:
113     printDegrees(Root, adj);
(gdb) n
1: 3
2: 2
3: 0
4: 1
5: 0
6: 0
7: 0
115     return 0;
(gdb) n
116 }
(gdb) n
```