

IT003.P21.CTTN - Assignment 04.01

Bảo Quý Định Tân - 24520028

Hà Xuân Thiện - 24520031

Đề bài

- a. Viết hàm chuyển cây BST sang Doubly Linked List tăng.
- b. Viết hàm chuyển Doubly Linked List tăng sang cây BST.
- **Lưu ý:** Không tạo thêm phần tử mới (tức không giữ lại cấu trúc dữ liệu cũ).

Code

Tham khảo chatgpt

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 struct Node
6 {
7     int data;
8     Node *left;
9     Node *right;
10
11     Node(int v) : data(v), left(nullptr), right(nullptr) {}
12 };
13
14 void insertBST(Node *&root, int val)
15 {
16     if (!root)
17     {
18         root = new Node(val);
19         return;
20     }
21
22     if (val < root->data)
23     {
24         insertBST(root->left, val);
25     }
26     else
27     {
28         insertBST(root->right, val);
```

```

29     }
30 }
31
32 void printBSTInOrder(Node *root)
33 {
34     if (!root)
35         return;
36
37     queue<Node *> q;
38     q.push(root);
39
40     cout << "Xuat BST ra theo tung tang:\n";
41     while (!q.empty())
42     {
43         int currLevelSize = q.size();
44         while (currLevelSize--)
45         {
46             Node *curr = q.front();
47             q.pop();
48
49             cout << curr->data << ' ';
50
51             if (curr->left)
52             {
53                 q.push(curr->left);
54             }
55             if (curr->right)
56             {
57                 q.push(curr->right);
58             }
59         }
60         cout << '\n';
61     }
62 }
63
64 Node *tmpPrevNode = nullptr;
65 void BST2DLL(Node *BSTRoot, Node *&DLLNode)
66 {
67     if (!BSTRoot)
68     {
69         return;
70     }
71
72     BST2DLL(BSTRoot->left, DLLNode);
73
74     if (!tmpPrevNode)
75     {
76         DLLNode = BSTRoot;
77     }
78     else
79     {

```

```

80     tmpPrevNode->right = BSTRoot;
81     BSTRoot->left = tmpPrevNode;
82 }
83 tmpPrevNode = BSTRoot;
84
85 BST2DLL(BSTRoot->right, DLLNode);
86 }
87
88 void printDLL(Node *DLLNode)
89 {
90     cout << "Doubly_Linked_List: ";
91     while (DLLNode)
92     {
93         cout << DLLNode->data << " ";
94         DLLNode = DLLNode->right;
95     }
96     cout << '\n';
97 }
98
99 Node *DLL2BST(Node *&DLLNode, int n)
100 {
101     if (!n || !DLLNode)
102     {
103         return nullptr;
104     }
105
106     Node *leftNode = DLL2BST(DLLNode, n / 2);
107
108     Node *currNode = DLLNode;
109     currNode->left = leftNode;
110
111     DLLNode = DLLNode->right;
112     currNode->right = DLL2BST(DLLNode, n - n / 2 - 1);
113
114     return currNode;
115 }
116
117 signed
118 main()
119 {
120     ios_base::sync_with_stdio(false);
121     cin.tie(nullptr);
122
123     int arr[] = {1, 10, 5, 1, 7, 15, 12, 20};
124     int n = sizeof(arr) / sizeof(arr[0]);
125
126     Node *BSTRoot = nullptr;
127
128     // Tao BST
129     for (int i = 0; i < n; i++)
130     {

```

```

131     insertBST(BSTRoot, arr[i]);
132 }
133
134 // Xuat ra de xem BST
135 printBSTInOrder(BSTRoot);
136
137 // Chuyen tu BST sang DLL tang
138 Node *DLLNode = nullptr;
139 BST2DLL(BSTRoot, DLLNode);
140 printDLL(DLLNode);
141
142 // Chuyen tu DLL tang sang BST
143 Node *newBSTRoot = DLL2BST(DLLNode, n);
144 printBSTInOrder(newBSTRoot);
145
146 return 0;
147 }
148
149 /*
150
151
152 Wednesday, 09 April 2025
153 12:09:04
154 */

```

a. Chuyển BST sang DLL

Ta duyệt qua BST theo thứ tự giá trị các số tăng dần. Tại node lá đầu tiên của BST ta đi tới, cũng là node có giá trị nhỏ nhất, ta gán lại node này là node đầu tiên của DLL tăng của mình.

Rồi cứ như vậy lần lượt vừa duyệt vừa duy trì thêm một con trỏ prevNode để nối lại các node của BST (theo thứ tự giá trị tăng dần) thành DLL tăng của mình.

Ta chỉ chỉnh lại các kết nối của các node trong BST để nó trở thành một DLL tăng như ý muốn nên ta không cần giữ lại cấu trúc dữ liệu cũ.

Code

```

1 Node *tmpPrevNode = nullptr;
2 void BST2DLL(Node *BSTRoot, Node *&DLLNode)
3 {
4     if (!BSTRoot)
5     {
6         return;
7     }
8
9     BST2DLL(BSTRoot->left, DLLNode);
10
11     if (!tmpPrevNode)

```

```

12 {
13     DLLNode = BSTRoot;
14 }
15 else
16 {
17     tmpPrevNode->right = BSTRoot;
18     BSTRoot->left = tmpPrevNode;
19 }
20 tmpPrevNode = BSTRoot;
21
22 BST2DLL(BSTRoot->right, DLLNode);
23 }

```

b. Chuyển DLL thành BST

Ý tưởng để thực hiện chuyển từ DLL thành một cây BST (cân bằng) khá hay. Ta duyệt đệ quy với ý nghĩa là tạo ra cây BST gồm n node bắt đầu từ phần tử DLLNode hiện tại.

Đầu tiên ta sẽ cố tạo ra cây BST từ $n/2$ phần tử bên trái trước trong n phần tử hiện tại, rồi nối nó vào phần tử hiện tại vào nhánh bên trái của nó.

Sau đó ta tạo ra cây BST từ $n - n/2 - 1$ node tiếp theo trong DLL, rồi nối nó vào nhánh bên phải của node ở giữa hiện tại.

Vì ta cố gắng khiến cho hai nhánh có kích thước gần bằng nhau nên cây BST tạo ra sẽ gần cân bằng.

Code

```

1 Node *DLL2BST(Node *&DLLNode, int n)
2 {
3     if (!n || !DLLNode)
4     {
5         return nullptr;
6     }
7
8     Node *leftNode = DLL2BST(DLLNode, n / 2);
9
10    Node *currNode = DLLNode;
11    currNode->left = leftNode;
12
13    DLLNode = DLLNode->right;
14    currNode->right = DLL2BST(DLLNode, n - n / 2 - 1);
15
16    return currNode;
17 }

```