作业 3

陈俊铭 3210300364 （信息与计算科学）

2022 年 10 月 28 日

1 题目要求

作业 4.37：Write a function that takes as input a binary search tree, T, and two keys, k_1 and k_2, which are ordered so that $k_1 <= k_2$; and prints all elements X in the tree such that $k_1 <= Key(X) <= k_2$. Do not assume any information about the type of keys except that they can be ordered (consistently). Your program should run in O(K + log N) average time, where K is the number of keys printed. Bound the running time of your algorithm.

2 设计思路

代码（题目中的 BST 认为就是 AVL Tree）：

```
#include <iostream>
#include <vector>
#include <ctime>
#include <math.h>
#include "AvlTree.h"

using namespace std;

int main() {

    clock_t start, finish;

    //When k <= log (n)
    AvlTree<int> T0;
    vector<int> vec0 {100, 890, 33, 0};
    for(int i = 0; i < vec0.size(); ++i)
    {
        T0.insert(vec0[i]);
        cout << vec0[i] << " ";
    }
```

```
cout << endl;
T0.printTree();
cout << endl;

//When k = n
AvlTree<int> T1;
vector<int> vec1 {1000, 10000, 100000, 1000000};

int count1 = 0;
while(count1 < vec1.size())
{
    int n = vec1[count1];
    int k = n;


    for(size_t j{1}; j <= n; ++j)
    {
        T1.insert(j);
    }
    start = clock();
    T1.printElement(1,k);
    finish = clock();

    cout << "n = " << n << endl;
    cout << "Runtime: " << double(finish-start) / CLOCKS_PER_SEC << " s" << endl;

    count1++;
}

cout << endl;

//When k = log(n)
AvlTree<int> T2;
vector<int> vec2 {int(pow(2,14)), int(pow(2,16)), int(pow(2,18)), int(pow(2,20))};

int count2 = 0;
while(count2 < vec2.size())
{
    int n = vec2[count2];
    int k = log2(n);


    for(size_t j{0}; j <= n; ++j)
    {
        T2.insert(j);
    }

    start = clock();
    T2.printElement(n,n + k - 1);
    finish = clock();
```

```
        cout << "n =" << n << endl;
        cout << "Runtime: " << (finish-start) / CLOCKS_PER_SEC << " s" << endl;

        count2++;
    }


    return 0;
}
```

3 理论分析

程序的复杂度取决于 k:

- 当 $k <= log(n)$ 时为最佳案，复杂性为 $O(log(n))$.
- 当 $k = n$ 时为最差案，复杂性为 $O(N)$.
- 当 $k = log(n)$ 时为平均案，复杂性为 $O(k + log(n))$.

4 数值结果分析

当 $k <= log(n)$:

```
100 890 33 0
0 33 100 890
```

当 $k = n$:

```
n = 1000
Runtime: 0 s
n = 10000
Runtime: 0 s
n = 100000
Runtime: 0 s
n = 1000000
Runtime: 0 s
```

当 $k = log(n)$:

```
n =16384
Runtime: 0 s
n =65536
Runtime: 0 s
n =262144
Runtime: 0 s
n =1048576
Runtime: 0 s
```