

# 作业 2: Binary Search Tree

陈科辉 Keiver Pabula

19 Octobe,2022

作业 2: 二叉搜索排序算法。

## 1 设计思路

根据老师所发的头文件添加两种排序函数，一个是先乱序再排序；另一个则是直接排序

1. 创建相应的头文件
2. 添加 sortTree 用于有序的打印树
3. 在 main.cpp 中添加 BSTSorting 制作一个函数，在这个模板函数中，应用传入的数组，并传入模板中；先将异常情况排除；如果 `__mode = 0` 则正常运行，如果 `__mode = 1` 则先乱序用 `random_shuffle`
4. 生成树，然后再排序；打印排序前后的 tree，和所用的时间
5. 函数中使用 `clock()` 来计算排序时间，并求出其效率
6. 设计主函数：主函数只要用于输入 test 几次，数组的 size（大小）和排序的模式。注意数组的内容是由 size 所固定的，数组的元素是由 0-size 这个方法来得到的。
7. debugging 测试
8. 得到结果
9. 制作报告

## 2 测试说明

输出的内容:

2 次测试分别是当它是模式 0 和 1 时, 100 个数据的情况:

```
student@student-VirtualBox:~/homework/pr2/bst$ g++ -o run main.cpp
student@student-VirtualBox:~/homework/pr2/bst$ ./run
2
100 0
原来的的数组: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
6 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
94 95 96 97 98 99
排序后的数组: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
6 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
94 95 96 97 98 99
The sorting process spent: 0.000345
100 1
原来的的数组: 92 51 11 69 24 35 17 36 26 98 67 39 83 2 75 56 59 18 32 40 91 86 57 12 14 42 27 62 63 58 30 96 13 68 3 87 71 64 9 22 66 80 20 7
89 81 73 0 94 41 88 28 52 43 16 60 49 7 84 72 29 61 6 45 53 76 8 33 37 15 25 55 70 31 82 47 48 10 90 34 23 74 77 19 85 44 93 54 97 1 38 95 4
99 5 78 65 50 46
排序后的数组: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
6 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
94 95 96 97 98 99
The sorting process spent: 0.000219
```

4 次测试分别是当它是模式 0 和 1 时, 1000 和 10000 个数据的情况:

```
student@student-VirtualBox:~/homework/pr2/bst$ g++ -o run main.cpp
student@student-VirtualBox:~/homework/pr2/bst$ ./run
4
1000 0
The sorting process spent: 0.007642
1000 1
The sorting process spent: 0.000711
10000 1
The sorting process spent: 0.006959
10000 0
The sorting process spent: 0.686545
```

1 次测试, 如果模式错误自动退出:

```
student@student-VirtualBox:~/homework/pr2/bst$ ./run
2
1000 2
Wrong input, please enter 0 or 1
```

可以看到排序一切正常, 且可以得到他们各自的效率但是可以看到他们的效率并不稳定:

```
student@student-VirtualBox:~/homework/pr2/bst$ g++ -o run main.cpp
student@student-VirtualBox:~/homework/pr2/bst$ ./run
10
1000 0
The sorting process spent: 0.009109
1000 1
The sorting process spent: 0.000841
1000 0
The sorting process spent: 0.006149
1000 1
The sorting process spent: 0.000425
1000 0
The sorting process spent: 0.008225
1000 0
The sorting process spent: 0.010077
1000 0
The sorting process spent: 0.006069
1000 1
The sorting process spent: 0.000431
1000 1
The sorting process spent: 0.000475
1000 1
The sorting process spent: 0.000827
```