

## Assignment 4(Binary Search Trees) Report

### 1. Test Cases

'Skewed order data set'과 'Random order data set'으로 나누어서 각각 Search Tree와 AVL Tree에 실험을 진행했다. Skewed order의 경우 Tree의 size에 따라 '1, 2, 3, 4, 5, ... , size'와 같이 설정했으며, Random order는 Skewed order과 같이 오름차순으로 정렬된 경우를 Random 값을 이용하여 index를 swap하는 방식으로 설정하였다. 즉, 두 test set을 이루는 Entry의 집합은 같다.

### 2. Results

실험을 3번 진행하여 얻은 결과의 평균을 표로 정리하면 아래와 같다. 진행한 실험에 대한 결과 중 하나의 예시는 보고서 마지막 부분인 '5. Appendix'에 제시하겠다. Skewed와 Random의 각 data set은 같고, Tree의 종류만 다르다. Skewed Order에서, 'Size=1,000,000'인 경우는 코드가 정상적으로 실행되지 않아 'Size=100,000'의 경우까지 비교하도록 하겠다. (정상적으로 실행되지 않는 이유는 메모리 문제일 것으로 생각한다.)

#### 2-1) Insert Operation

Tree (Order) Size	Search Tree (Skewed)	AVL Tree (Skewed)	Search Tree (Random)	AVL Tree (Random)
10	0.000014	0.000037	0.000002	0.000014
100	0.000402	0.0004833	0.00003633	0.000111667
1,000	0.0073397	0.001343	0.0003097	0.0010323333
10,000	0.7045133	0.0160403	0.003814	0.013194667
100,000	85.295929	0.192861	0.0525683	0.165183
1,000,000	-	-	0.7202583	2.230299

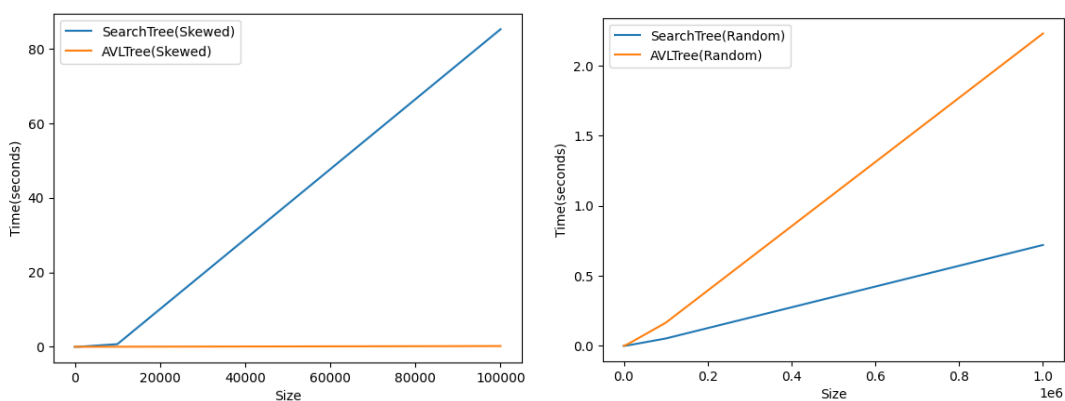
#### 2-2) Find Operation

Tree (Order) Size	Search Tree (Skewed)	AVL Tree (Skewed)	Search Tree (Random)	AVL Tree (Random)
10	0.00001	0.000007	0.00000633	0.000001
100	0.0003587	0.000062	0.000019	0.0000273
1,000	0.007031	0.0002057	0.0002967	0.000236
10,000	0.6946097	0.0020963	0.003333	0.002294
100,000	84.527934	0.0260117	0.0468427	0.02979533
1,000,000	-	-	0.6599847	0.48491767

### 3. Performance Comparison

#### 3-1) Comparison by size and type of test set

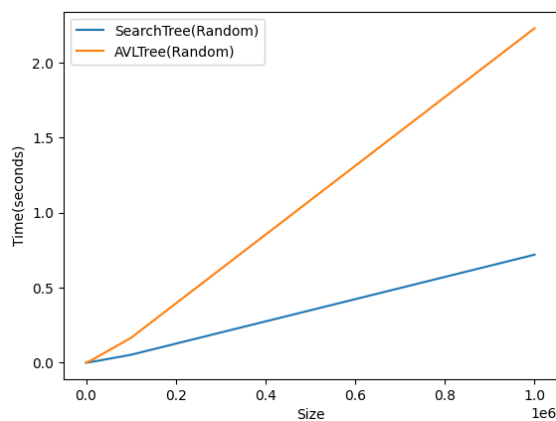
실험의 결과를 정리한 표를 통해 test set의 size가 커질수록 operation에 필요한 시간이 늘어나는 것을 확인할 수 있다. 또한, Skewed인 경우가 Random의 경우보다 더 오래 걸리는 것을 확인할 수 있다. 특히, Search Tree의 경우, size가 커질수록 Operation에 걸리는 시간이 급격하게 증가하는 것을 확인할 수 있다. AVL Tree의 경우, Skewed data set에 대하여 height balance를 위한 rotation 연산이 많아질 것으로 생각할 수 있다. Insert Operation의 경우를 그래프로 그리면 아래와 같다.



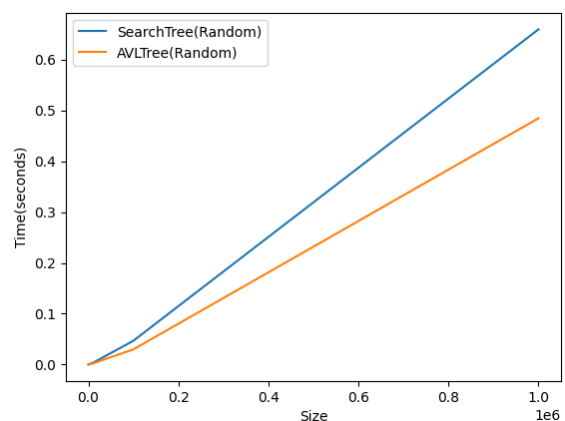
#### 3-2) Comparison by type of tree

Tree의 일반적인 경우에 대하여 분석하기 위하여 Random data set인 경우에 대하여 비교하겠다. AVL Tree는 height balanced tree이므로, Insert Operation의 과정에서 height를 조건에 만족시키기 위한 Rotation 연산이 Search Tree의 경우보다 추가적으로 요구된다. 따라서, 아래 좌측 그래프를 보면, Insert Operation의 경우 AVL Tree의 수행시간이 더 오래 걸리는 것을 확인할 수 있다. 그러나, Find Operation의 경우, 두 Tree 모두  $O(\text{height})$ 의 수행 시간이 기대된다. 따라서, 아래 우측 그래프를 보면, AVL Tree의 경우가 수행시간이 더 적음을 알 수 있다.

Insert Operation



Find Operation



## 4. Conclusions

두 가지 Tree와 두 가지 data set에 대하여 실험을 진행하였다. C++ 언어 구현을 하면서, class 간의 접근과 함수 사용에 대하여 좋은 경험을 할 수 있는 과제였다. 무엇보다 수업 시간에 배웠던 개념을 직접 구현해보는 것이 좋았다. 평소  $O(\log n)$ 과  $O(n)$ 의 차이가 별로 크지 않을 것 같다고 생각했는데, 이번 실험 과제를 통하여 둘의 차이를 체감한 것 같다. Tree의 종류에 따라, Operation의 종류에 따라, 그리고 Tree를 이루는 Entry의 특성에 따라 기대되는 수행 시간이 다르기 때문에 주어진 상황에 맞춰 어떤 것이 적절할지 판단해야 될 것 같다.

## 5. Appendix

### 5-1) Running Given Skeleton Code and Example of Experiments

```
p2016320130@hvcl:~$ make
g++ main.cpp -o assign_4
p2016320130@hvcl:~$ ./assign_4
---Let's test given skeleton code---
Initial Size : 0
Size before insert same key 3 : 4
Size after insert same key 3 : 4
Size is same

Tree has below entries
Size : 4
Key : 1, Value : 2.5
Key : 2, Value : 1.5
Key : 3, Value : 8.5
Key : 7, Value : 5.5

Size before erase key 3 : 4
Size after erase key 3 : 3

Tree has below entries
Size : 3
1 , 2.5
2 , 1.5
7 , 5.5
---Given skeleton code works well, let's start experiment---
```

```
Test for the size of 10

Insert SearchTree by using skewed data set
It took me 0.000003 seconds.
Find SearchTree by using skewed data set
It took me 0.000002 seconds.
Insert AVLTree by using skewed data set
It took me 0.000009 seconds.
Find AVL tree by using skewed data set
It took me 0.000002 seconds.

Test for the size of 100

Insert SearchTree by using skewed data set
It took me 0.000125 seconds.
Find SearchTree by using skewed data set
It took me 0.000100 seconds.
Insert AVLTree by using skewed data set
It took me 0.000153 seconds.
Find AVL tree by using skewed data set
It took me 0.000028 seconds.

Test for the size of 1000

Insert SearchTree by using skewed data set
It took me 0.007476 seconds.
Find SearchTree by using skewed data set
It took me 0.007370 seconds.
Insert AVLTree by using skewed data set
It took me 0.001427 seconds.
Find AVL tree by using skewed data set
It took me 0.000209 seconds.

Test for the size of 10000

Insert SearchTree by using skewed data set
It took me 0.751858 seconds.
Find SearchTree by using skewed data set
It took me 0.722159 seconds.
Insert AVLTree by using skewed data set
It took me 0.016624 seconds.
Find AVL tree by using skewed data set
It took me 0.002268 seconds.
```

Test for the size of 100000

Insert SearchTree by using skewed data set  
It took me 102.975319 seconds.  
Find SearchTree by using skewed data set  
It took me 90.042648 seconds.  
Insert AVLTree by using skewed data set  
It took me 0.192113 seconds.  
Find AVL tree by using skewed data set  
It took me 0.026097 seconds.

Test for the size of 10

Insert SearchTree by using random data set  
It took me 0.000002 seconds.  
Find SearchTree by using random data set  
It took me 0.000001 seconds.  
Insert AVLTree by using random data set  
It took me 0.000003 seconds.  
Find AVLTree by using random data set  
It took me 0.000001 seconds.

Test for the size of 100

Insert SearchTree by using random data set  
It took me 0.000048 seconds.  
Find SearchTree by using random data set  
It took me 0.000015 seconds.  
Insert AVLTree by using random data set  
It took me 0.000109 seconds.  
Find AVLTree by using random data set  
It took me 0.000026 seconds.

Test for the size of 1000

Insert SearchTree by using random data set  
It took me 0.000362 seconds.  
Find SearchTree by using random data set  
It took me 0.000335 seconds.  
Insert AVLTree by using random data set  
It took me 0.000989 seconds.  
Find AVLTree by using random data set  
It took me 0.000229 seconds.

Test for the size of 1000

Insert SearchTree by using random data set  
It took me 0.000362 seconds.  
Find SearchTree by using random data set  
It took me 0.000335 seconds.  
Insert AVLTree by using random data set  
It took me 0.000989 seconds.  
Find AVLTree by using random data set  
It took me 0.000229 seconds.

Test for the size of 10000

Insert SearchTree by using random data set  
It took me 0.003424 seconds.  
Find SearchTree by using random data set  
It took me 0.002941 seconds.  
Insert AVLTree by using random data set  
It took me 0.012707 seconds.  
Find AVLTree by using random data set  
It took me 0.002283 seconds.

Test for the size of 100000

Insert SearchTree by using random data set  
It took me 0.046319 seconds.  
Find SearchTree by using random data set  
It took me 0.041328 seconds.  
Insert AVLTree by using random data set  
It took me 0.166361 seconds.  
Find AVLTree by using random data set  
It took me 0.029696 seconds.

Test for the size of 1000000

Insert SearchTree by using random data set  
It took me 0.708064 seconds.  
Find SearchTree by using random data set  
It took me 0.650110 seconds.  
Insert AVLTree by using random data set  
It took me 2.226845 seconds.  
Find AVLTree by using random data set  
It took me 0.483636 seconds.  
p2016320130@hvcl:~\$ |