GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 7 Report

Oğuzhan SEZGİN
1801042005

Q2

Only the insert operation of the myskiplist class is running.

Q3

Average times for the elements added and deleted in the trees are as follows.

Tree size : 10.000

Add operation

Binary search tree add average(10000) time : 8460 ns
Red Black Tree add average(10000) time : 7570 ns
Skiplist(book) add average(10000) time : 12920 ns
Skiplist(java) add average(10000) time : 14090 ns
BTree add average(10000) time : 7220 ns
Skiplist(mine) add average(10000) time : 10550 ns

Remove Operation

Binary search tree remove average(10000) time : 7680 ns
Red Black Tree remove average(10000) time : 4960 ns
Skiplist(book) remove average(10000) time : 6610 ns
Skiplist(java) remove average(10000) time : 29070 ns

Tree size : 20.000

Add operation

Binary search tree add average(20000) time : 10920 ns
Red Black Tree add average(20000) time : 8960 ns
Skiplist(book) add average(20000) time : 18310 ns
Skiplist(java) add average(20000) time : 18380 ns
BTree add average(20000) time : 9450 ns
Skiplist(mine) add average(20000) time : 23230 ns

Remove Operation

Binary search tree remove average(20000) time : 8750 ns
Red Black Tree remove average(20000) time : 13250 ns
Skiplist(book) remove average(20000) time : 9650 ns
Skiplist(java) remove average(20000) time : 44070 ns

Tree size : 40.000

Add operation

Binary search tree add average(40000) time : 10900 ns
Red Black Tree add average(40000) time : 8930 ns
Skiplist(book) add average(40000) time : 15470 ns
Skiplist(java) add average(40000) time : 18360 ns
BTree add average(40000) time : 8990 ns
Skiplist(mine) add average(40000) time : 12440 ns

Remove Operation

Binary search tree remove average(40000) time : 11050 ns
Red Black Tree remove average(40000) time : 6890 ns
Skiplist(book) remove average(40000) time : 8350 ns
Skiplist(java) remove average(40000) time : 35450 ns

Tree size : 80.000

Add operation

Binary search tree add average(80000) time : 11310 ns
Red Black Tree add average(80000) time : 11550 ns

Skiplist(book) add average(80000) time : 16360 ns
Skiplist(java) add average(80000) time : 20570 ns
BTree add average(80000) time : 13650 ns
Skiplist(mine) add average(80000) time : 33320 ns

Remove Operation

Binary search tree remove average(80000) time : 7590 ns
Red Black Tree remove average(80000) time : 4980 ns
Skiplist(book) remove average(80000) time : 8920 ns
Skiplist(java) remove average(80000) time : 28700 ns

According to the above results, Btree performs best. Then the Red-Black tree comes after the binary search tree followed by the skiplist in book followed by the skiplist in java and finally the skiplist I wrote for q2.