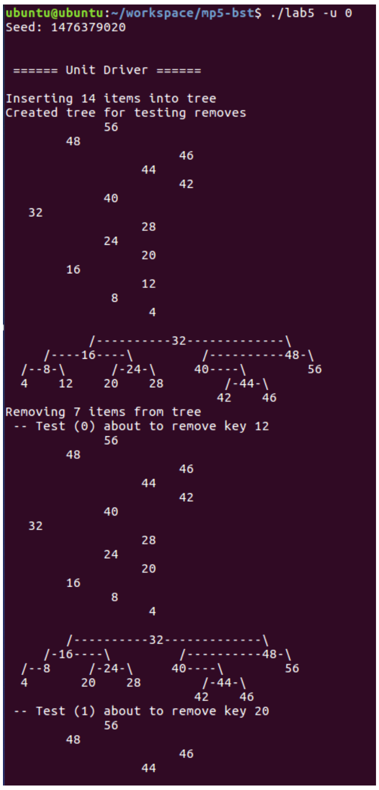
Test Log For MP5

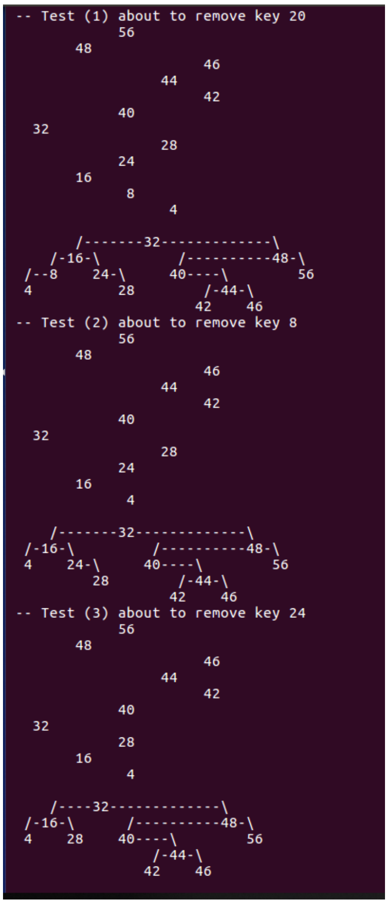
Student: Yulin Xiao
CUID: C16278133
Student UserName: Yulinx

1.Do the unit driver with custom test 0 through 6 (4-6 is my added binary search tree tests) to verify that my code is able to do the basic binary search tree operation (insert, remove):

```
lab5 -u 0
//test to remove leaves, 12 and 20, then internal nodes
// 8, 24, 40 with one child, then 16, 48 with two children
```



```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -u 0
Seed: 1476379020

 ====== Unit Driver ======

Inserting 14 items into tree
Created tree for testing removes
              56
         48
                        46
                   44
                        42
              40
    32
                   28
              24
                   20
         16
                   12
              8
                   4

            /----------32------------\
      /----16----\           /----------48-\
   /--8-\      /-24-\     40----\          56
   4    12    20    28         /-44-\
                              42    46
Removing 7 items from tree
 -- Test (0) about to remove key 12
              56
         48
                        46
                   44
                        42
              40
    32
                   28
              24
                   20
         16
              8
                   4

         /---------32------------\
      /-16----\          /----------48-\
   /--8      /-24-\    40----\          56
   4        20    28        /-44-\
                           42    46
 -- Test (1) about to remove key 20
              56
         48
                        46
                   44
```



```
 -- Test (1) about to remove key 20
              56
         48
                        46
                   44
                        42
              40
    32
                   28
              24
         16
              8
                   4

         /-------32-------------\
      /-16-\           /----------48-\
   /--8    24-\     40----\          56
   4          28         /-44-\
                        42    46
 -- Test (2) about to remove key 8
              56
         48
                        46
                   44
                        42
              40
    32
                   28
              24
         16
              4

      /-------32-------------\
   /-16-\          /----------48-\
   4    24-\     40----\          56
           28         /-44-\
                     42    46
 -- Test (3) about to remove key 24
              56
         48
                        46
                   44
                        42
              40
    32
              28
         16
              4

      /----32-------------\
   /-16-\       /----------48-\
   4    28    40----\          56
                   /-44-\
                  42    46
```

```
-- Test (4) about to remove key 40
            56
       48
                 46
            44
                 42
  32
            28
       16
             4

    /----32----------\
 /-16-\         /----48-\
 4    28      /-44-\     56
            42    46
-- Test (5) about to remove key 16
            56
       48
                 46
            44
                 42
  32
            28
             4

    /-32----------\
 /-28        /----48-\
 4        /-44-\     56
         42    46
-- Test (6) about to remove key 48
       56
                 46
            44
                 42
  32
       28
             4

    /-32----------\
 /-28        /----56
 4        /-44-\
         42    46
```

lab5 -u 1

// tests: (48) is missing its right-left child and

//   (16) is missing its left-right child

```
 ====== Unit Driver ======

Inserting 13 items into tree
Created tree for testing removes
                  60
             56
        48
                  44
             40
                  36
   32
                  28
             24
                  20
        16
              8
                   4

        /----------32----------\
     /-16----\            /----48-\
  /--8     /-24-\      /-40-\     56-\
  4       20    28    36    44        60
Removing 3 items from tree
 -- Test (0) about to remove key 16
                  60
             56
        48
                  44
             40
                  36
   32
                  28
             24
        20
              8
                   4

        /-------32----------\
     /-20-\            /----48-\
  /--8    24-\      /-40-\     56-\
  4          28    36    44       60
 -- Test (1) about to remove key 48
             60
        56
                  44
             40
                  36
   32
                  28
             24
        20
              8
                   4
```

```
 -- Test (1) about to remove key 48
                  60
             56
        52
             40
                  36
   32
             28
        24
                  12
              8
                   4

           /----32-------\
     /----24-\         /-52-\
  /--8-\     28     /-40    56-\
  4    12          36          60
 -- Test (2) about to remove key 32
                  60
             56
        52
             40
   36
             28
        24
                  12
              8
                   4

           /----36----\
     /----24-\      /-52-\
  /--8-\     28    40    56-\
  4    12                  60
```

lab5 -u 2

// deletion with many children

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -u 2
Seed: 1476379020


 ====== Unit Driver ======

Inserting 16 items into tree
Created tree for testing removes
  200
                 175
            150
                           140
                      135
                           130
                 125
       100
                       85
                                 82
                            80
                                 78
                  75
                       65
             50
                  25
                          /------------------200
     /------------------100-------------\
  /-50----\                    /---------150-\
 25      /-75----------\     125----\              175
       65         /----85          /135-\
              /-80-\              130    140
             78     82
Removing 3 items from tree
 -- Test (0) about to remove key 100
  200
                 175
            150
                      140
                 135
                      130
       125
                       85
                                 82
                            80
                                 78
                  75
                       65
             50
                  25
```

```
                         /--------------200
    /----------------125---------\
 /-50----\                      /---150-\
25     /-75---------\      /135-\    175
      65        /----85   130   140
             /-80-\
            78    82
-- Test (1) about to remove key 85
 200
                175
           150
                     140
                135
                     130
      125
                           82
                      80
                           78
                 75
                      65
            50
                 25

                      /--------------200
    /-------------125---------\
 /-50----\                  /---150-\
25     /-75----\         /135-\    175
      65     /-80-\    130   140
            78    82
-- Test (2) about to remove key 125
 200
                175
           150
                     140
                135
      130
                           82
                      80
                           78
                 75
                      65
            50
                 25

                      /-----------200
    /-------------130-------\
 /-50----\               /---150-\
25     /-75----\       135-\     175
      65     /-80-\       140
            78    82
```

lab5 -u 3

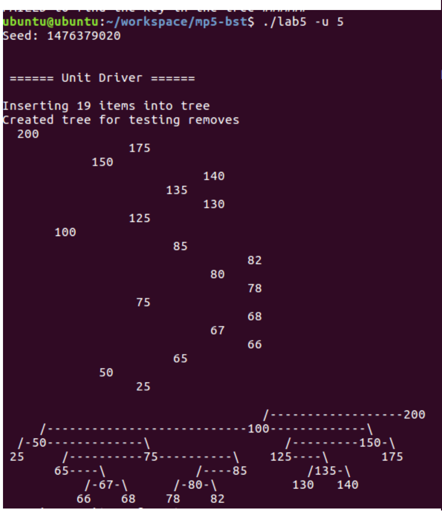
// check replace for duplicate key

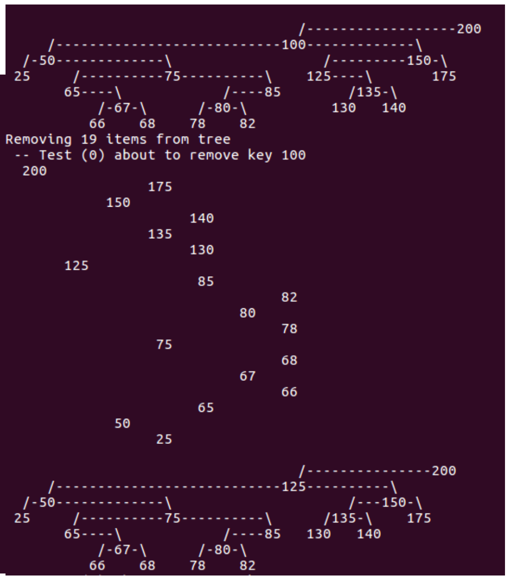
```
Seed: 1476379020


 ====== Unit Driver ======

Inserting 2 items into tree
Created tree for testing removes
   10

 10
Removing 1 items from tree
 -- Test (0) about to remove key 10
```

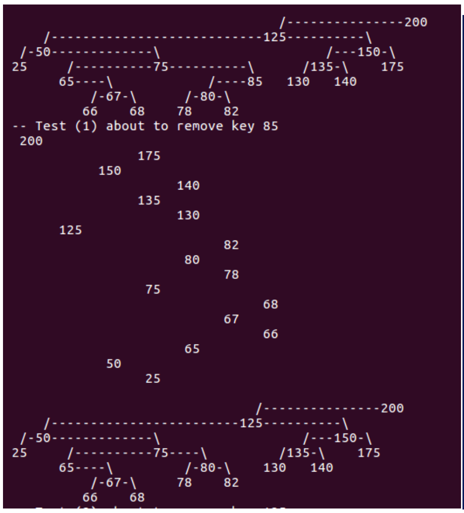
lab5 -u 4

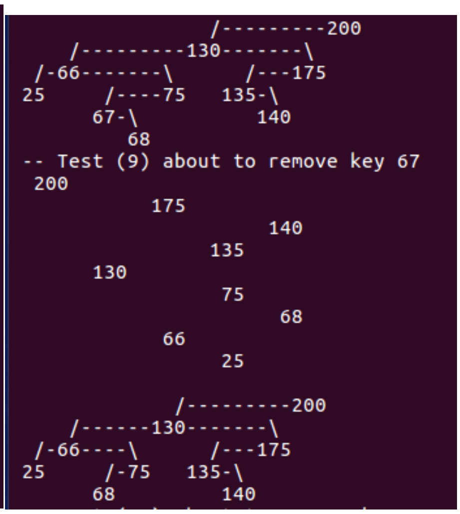
// check replace and double deletion for duplicate key lab5 -u 5

// complete deletion, first for parent(100,85(L),65(R),200(Root)) and then for child(67,68,66)

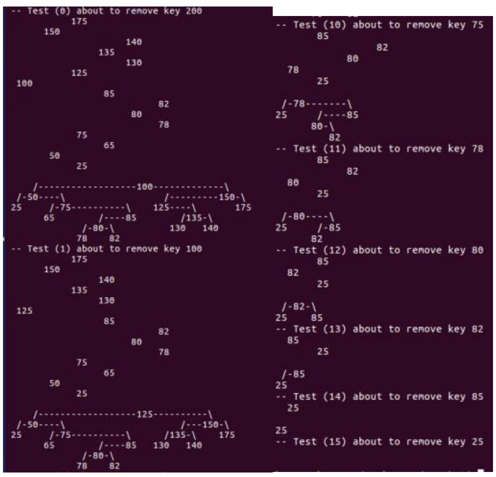




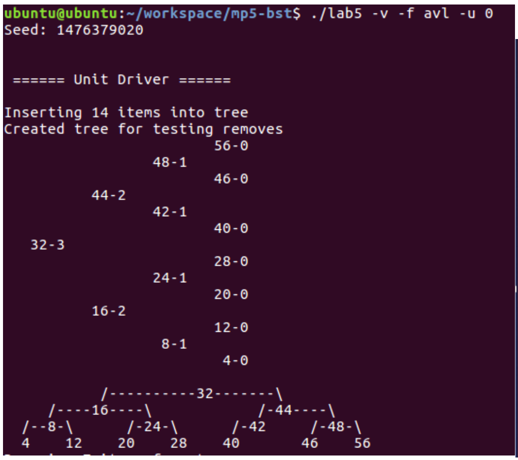


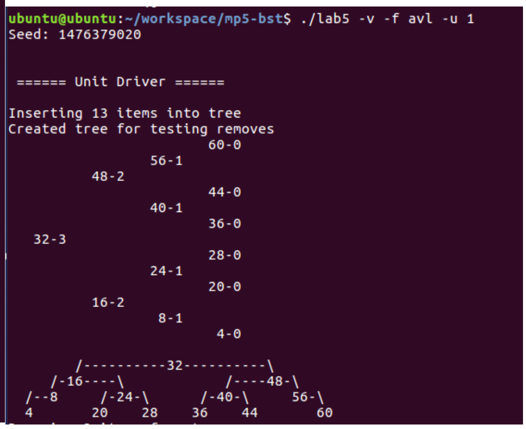


lab5 -u 6 // complete deletion, remove root



2. Do the unit driver 0 and 1 with ./lab5 -v -f avl to test my AVL tree insertion (the deletion is not completed, but the deletion do not violate the AVL property, so it do not generate the assertion fault) (Check the property)

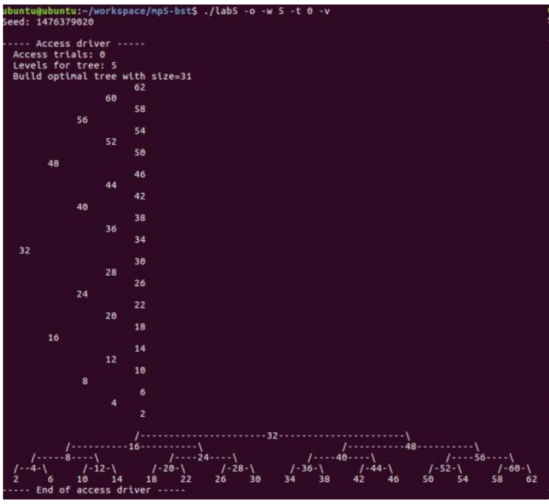


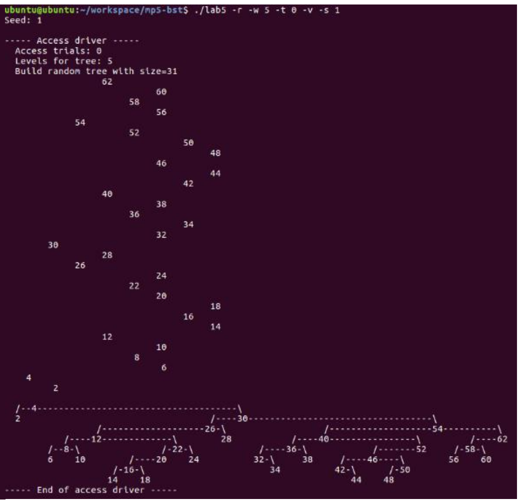


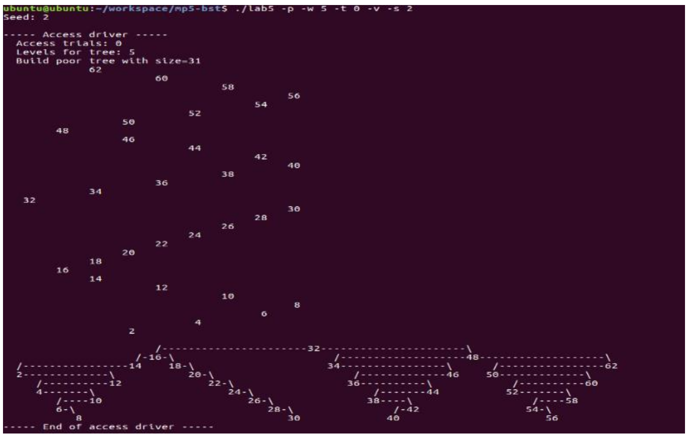
3. Do the command line arguments of:

```
lab5 -o -w 5 -t 0 -v
//  tests  inserts  only  and  prints  tree
lab5 -r -w 5 -t 0 -v -s 1
// same  with  random  tree
lab5 -p -w 5 -t 0 -v -s 2
// same  with  random  tree
```

To validate that the tree remains in binary search tree's property: 1) bst







2) avl:

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -o -w 5 -t 0 -v
Seed: 1476379020

----- Access driver -----
  Access trials: 0
  Levels for tree: 5
  Build optimal tree with size=31
                               62-0
                        60-1
                               58-0
                 56-2
                               54-0
                        52-1
                               50-0
          48-3
                               46-0
                        44-1
                               42-0
                 40-2
                               38-0
                        36-1
                               34-0
   32-4
                               30-0
                        28-1
                               26-0
                 24-2
                               22-0
                        20-1
                               18-0
          16-3
                               14-0
                        12-1
                               10-0
                  8-2
                                6-0
                         4-1
                                2-0

                         /---------------------32---------------------\
           /----------16----------\                        /----------48----------\
     /-----8----\              /----24----\            /----40----\              /----56----\
  /--4-\      /-12-\      /-20-\      /-28-\      /-36-\      /-44-\      /-52-\      /-60-\
  2     6    10    14    18    22    26    30    34    38    42    46    50    54    58    62
----- End of access driver -----
```

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -r -w 5 -t 0 -v -s 1
Seed: 1

----- Access driver -----
  Access trials: 0
  Levels for tree: 5
  Build random tree with size=31
                               62-1
                                      60-0
                        58-2
                                      56-0
                               54-1
                 52-3
                        50-1
                               48-0
          46-4
                               44-0
                        42-1
                 40-3
                                      38-0
                               36-1
                        34-2
                               32-0
   30-5
                               28-0
                        26-1
                               24-0
                 22-3
                               20-1
                                      18-0
                        16-2
                               14-0
          12-4
                               10-0
                         8-1
                                6-0
                  4-2
                         2-0

                 /-----------------------30---------------------\
     /----------12-------------\                             /-------46-------\
  /--4----\            /-------22----\            /-------40-\              /-52-------\
  2     /--8-\     /-16----\       /-26-\     /-34-\         42-\      /-50      /----58----\
        6    10    14     /-20    24    28    32    36-\        44    48       54-\      /-62
                         18                          38                         56    60
----- End of access driver -----
```

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -p -w 5 -t 0 -v -s 2
Seed: 2

----- Access driver -----
  Access trials: 0
  Levels for tree: 5
  Build poor tree with size=31
                                    62-0
                             60-1
                      58-2
                                    56-0
                             54-1
                                    52-0
               50-3
                             48-1
                                    46-0
                      44-2
                                    42-0
                             40-1
                                    38-0
        36-4
                      34-0
               32-2
                             30-0
                      28-1
 26-5
                      24-1
                             22-0
               20-2
                      18-1
                             16-0
        14-3
                             12-0
                      10-1
                              8-0
                6-2
                       4-1
                              2-0

                   /----------------26-------------\
      /----------14-------\                  /----36------------------\
    /--6----\          /-20----\       /----32-\              /-------50----------\
  /--4      /-10-\      /-18     /-24    28-\      34          /----44----\          /----58-\
  2         8     12    16        22        30               /-40-\     /-48      /-54-\    60-\
                                                             38    42    46        52    56      62
----- End of access driver -----
```

4. Do the command line arguments of:

```
lab5 -o -w 20 -t 1000000
// tests inserts and accesses
lab5 -r -w 20 -t 1000000
// same with random tree
lab5 -p -w 20 -t 1000000
// same with poor insertion order
```

To verify that the expected number of searches predicted by the theory matches the measured performance from my program to three significant digits when run with 1,000,000 trials. 1) bst:

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f bst -o -w 20 -t 1000000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000000
  Levels for tree: 20
  Build optimal tree with size=1048575
  After access exercise, time=977.116, tree size=1048575
    Expect successful search=37, measured=19, trials=499682
    Expect unsuccessful search=40, measured=19, trials=500318
----- End of access driver -----

ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f bst -r -w 20 -t 1000000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000000
  Levels for tree: 20
  Build random tree with size=1048575
  After access exercise, time=1384.18, tree size=1048575
    Expect successful search=49.9952, measured=28, trials=499871
    Expect unsuccessful search=52.9952, measured=28, trials=500129
----- End of access driver -----

ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f bst -p -w 20 -t 1000000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000000
  Levels for tree: 20
  Build poor tree with size=1048575
  After access exercise, time=5889.4, tree size=1048575
    Expect successful search=1042, measured=1032, trials=499703
    Expect unsuccessful search=1045, measured=1032, trials=500297
----- End of access driver -----
```

2) avl:

```
ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -p -w 10 -t 1000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000
  Levels for tree: 10
  Build poor tree with size=1023
  After access exercise, time=0.133, tree size=1023
    Expect successful search=17.4633, measured=11, trials=467
    Expect unsuccessful search=20.4434, measured=11, trials=533
----- End of access driver -----

ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -r -w 10 -t 1000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000
  Levels for tree: 10
  Build random tree with size=1023
  After access exercise, time=0.131, tree size=1023
    Expect successful search=17.4282, measured=10, trials=477
    Expect unsuccessful search=20.4082, measured=10, trials=523
----- End of access driver -----

ubuntu@ubuntu:~/workspace/mp5-bst$ ./lab5 -f avl -o -w 10 -t 1000
Seed: 1476379020

----- Access driver -----
  Access trials: 1000
  Levels for tree: 10
  Build optimal tree with size=1023
  After access exercise, time=0.185, tree size=1023
    Expect successful search=17.0196, measured=9, trials=471
    Expect unsuccessful search=20, measured=9, trials=529
----- End of access driver -----
```

bst: best: 37, 40

average: 50, 53

worst: 1042, 1045

avl: best: 17, 20

average: 17, 20

worst: 17, 20

5. I find out that in standish's textbook:

Case	C_n	$C_n{}'$
Best	$2log_2n - 3$	log_2n
Average	$2.77log_2n - 4.7$	$2.77log_2n - 1.7$
Worst	n	n + 2

And what I found: (bst->20 levels)

Case	$C_{2^{20}}$ (Mine)	$C_{2^{20}}{}'$(Mine)
Best	$2log_2 2^{20} - 3$ = 37 (37)	$2log_2 2^{20}$ = 40 (40)
Average	$2.77log_2 2^{20} - 4.7$ = 50.7 (50)	$2.77log_2 2^{20} - 1.7$ = 53.7(53)
Worst	2^{20} (1000000) (1042)	2^{20} + 2 (1000002) (1045)

And what I found: (avl->10 levels)

Case	$C_{2^{10}}$(Mine)	$C_{2^{10}}{}'$(Mine)
Best	$2log_2 2^{10} - 3$ = 17 (17)	$2log_2 2^{10}$ = 20 (20)
Average	$2.77log_2 2^{10} - 4.7$ = 23 (17)	$2.77log_2 2^{10} - 1.7$ = 26 (20)
Worst	2^{10} (1000) (17)	2^{20} + 2 (1002) (20)

1) So by comparation of my number of successful and unsuccessful searches, it just perfectly match the Approximate Average Number of Comparisons for successful and unsuccessful search , so that my binary search tree works well in three cases.
2) As for worst case, I guess its time of complexity is O($\sqrt{n}$), for it generates trees that only have two subtrees that looks like linked list. For the time complexity of remove, insert and access of a certain node in linked list is O(n), and the poor tree looks like ^, it cuts more than 2 times in every moves downward.
3) As for the avl tree because its balanced characteristic, so, no matter in what policy the node is inserted, the avl tree matches the best cases.