CSCB63 Tutorial 2 — AVL trees
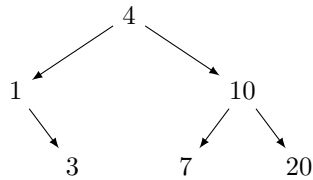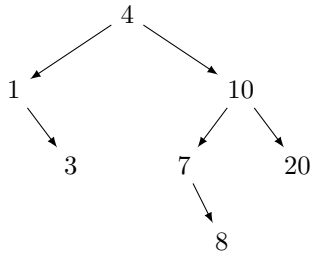
1 AVL tree: insert

Consider the following AVL tree.

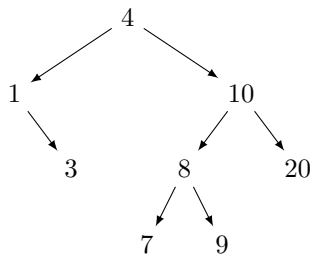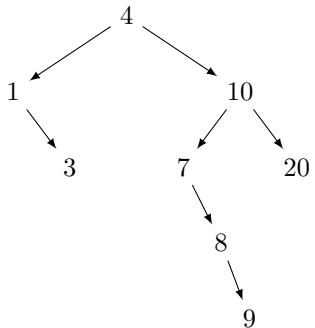

1. Give an example sequence of `insert` operations that, starting with an empty AVL tree, could have resulted in the above tree.
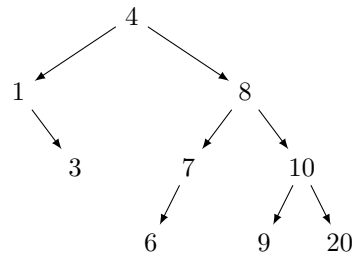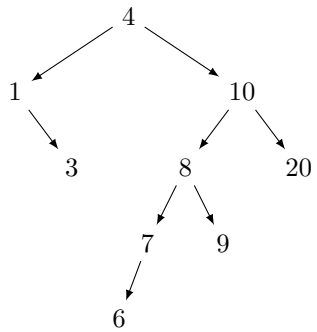
   4, 1, 10, 3, 7, 20

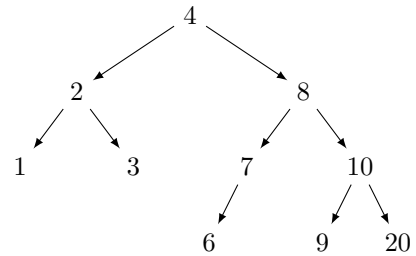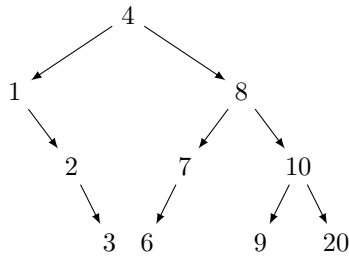2. Insert 8. Show the process and the resulting tree.

3. Insert 9 (into the tree resulting from inserting 8 above). Show the process and the resulting tree.

4. Insert 6 (into the tree resulting from inserting 9 above). Show the process and the resulting tree.

4
1 10
3 8 20
7 9
6

4
1 8
3 7 10
6 9 20

5. Insert 2 (into the tree resulting from inserting 6 above). Show the process and the resulting tree.

4
1 8
3 7 10
2 6 9 20

4
1 8
2 7 10
3 6 9 20

4
2 8
1 3 7 10
6 9 20

6. Finally, insert 22.

4
2 8
1 3 7 10
6 9 20
22
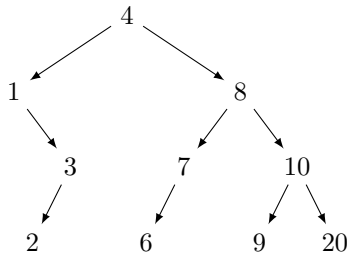
8
4 10
2 7 9 20
1 3 6 22
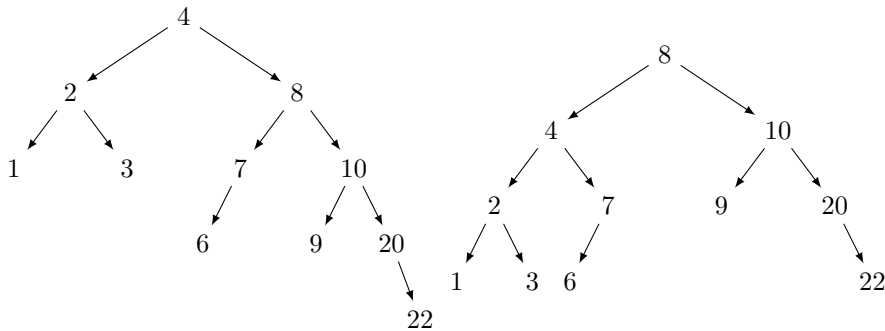
2 AVL tree: delete

We begin with the tree we constructed in the previous question.

8
4 10
2 7 9 20
1 3 6 22

1. Delete 9.

8
4 10
2 7 20
1 3 6 22

8
4 20
2 7 10 22
1 3 6

2

2. Delete 8.

```
            10
          ↙    ↘
        4        20
      ↙  ↘          ↘
     2     7          22
    ↙ ↘   ↙
   1   3 6
```

3. Delete 20.

```
       10                          4
     ↙    ↘                     ↙     ↘
    4       22                 2        10
  ↙  ↘                       ↙  ↘     ↙   ↘
 2     7                    1    3   7     22
↙ ↘   ↙                            ↙
1  3  6                            6
```
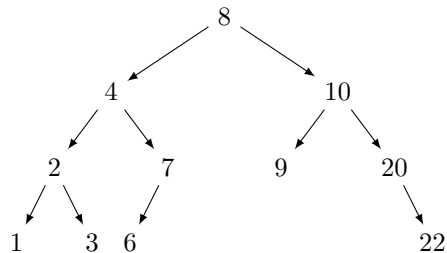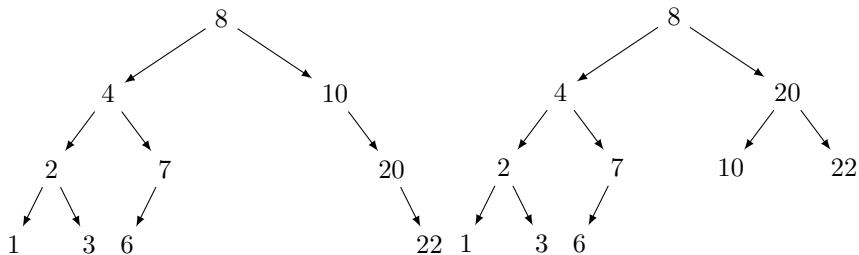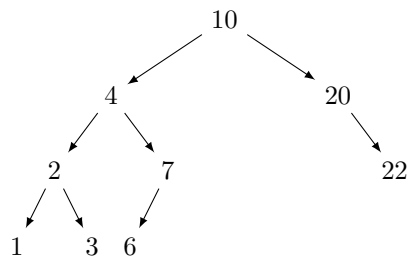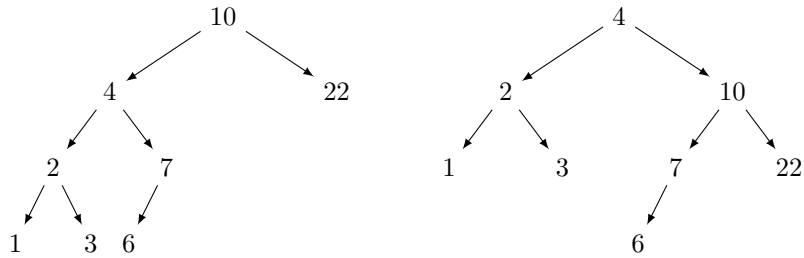
4. Now let's go back to the original tree, but with an added 0:

```
                8
              ↙    ↘
            4        10
          ↙  ↘      ↙  ↘
         2     7   9    20
        ↙ ↘   ↙             ↘
       1   3 6               22
      ↙
     0
```

Let's delete 8.

```
        9                              9
      ↙    ↘                         ↙    ↘
     4       10                     4       20
   ↙  ↘        ↘                  ↙  ↘     ↙  ↘
  2     7       20               2     7  10   22
 ↙ ↘   ↙          ↘             ↙ ↘   ↙
1   3  6           22          1   3  6
↙                             ↙
0                            0
```
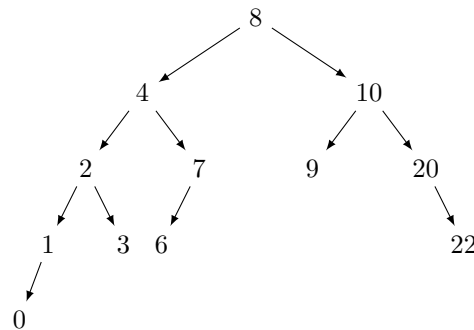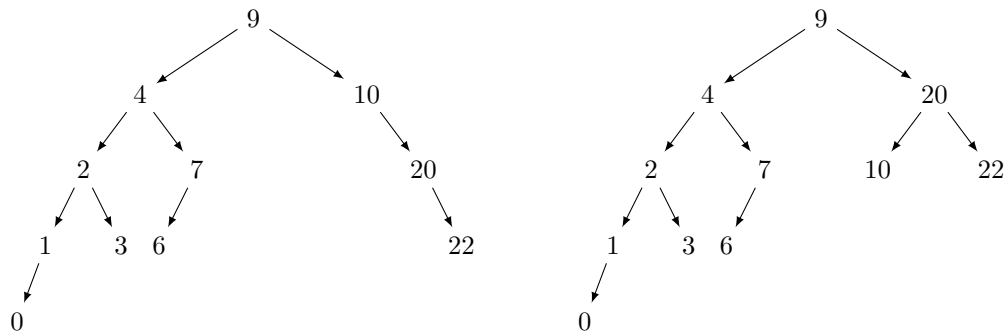
But this is not enough: the tree is still unbalanced. Need to rotate again.

```
                4
              ↙    ↘
            2        9
          ↙  ↘     ↙   ↘
         1    3   7     20
        ↙        ↙    ↙  ↘
       0        6   10    22
```