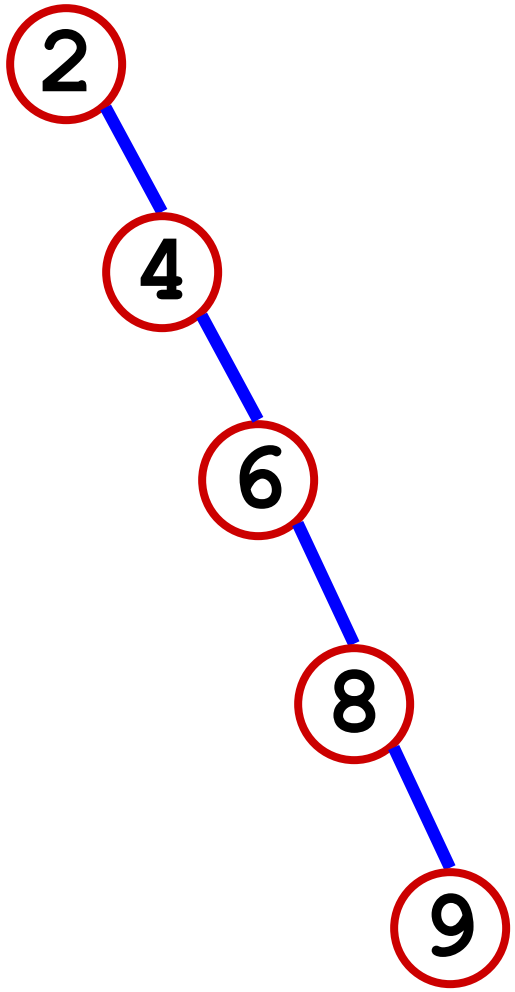
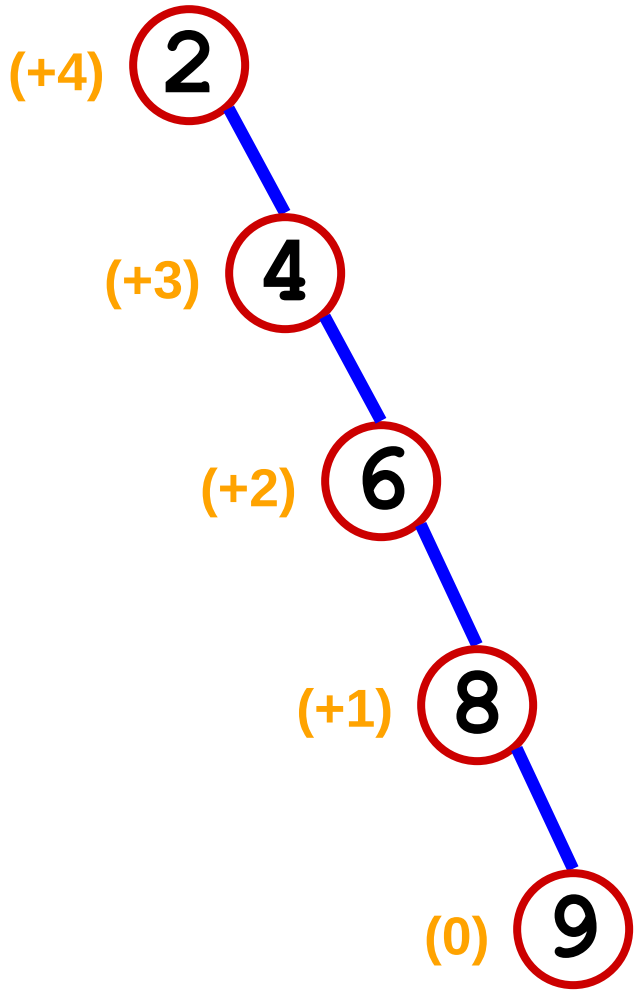
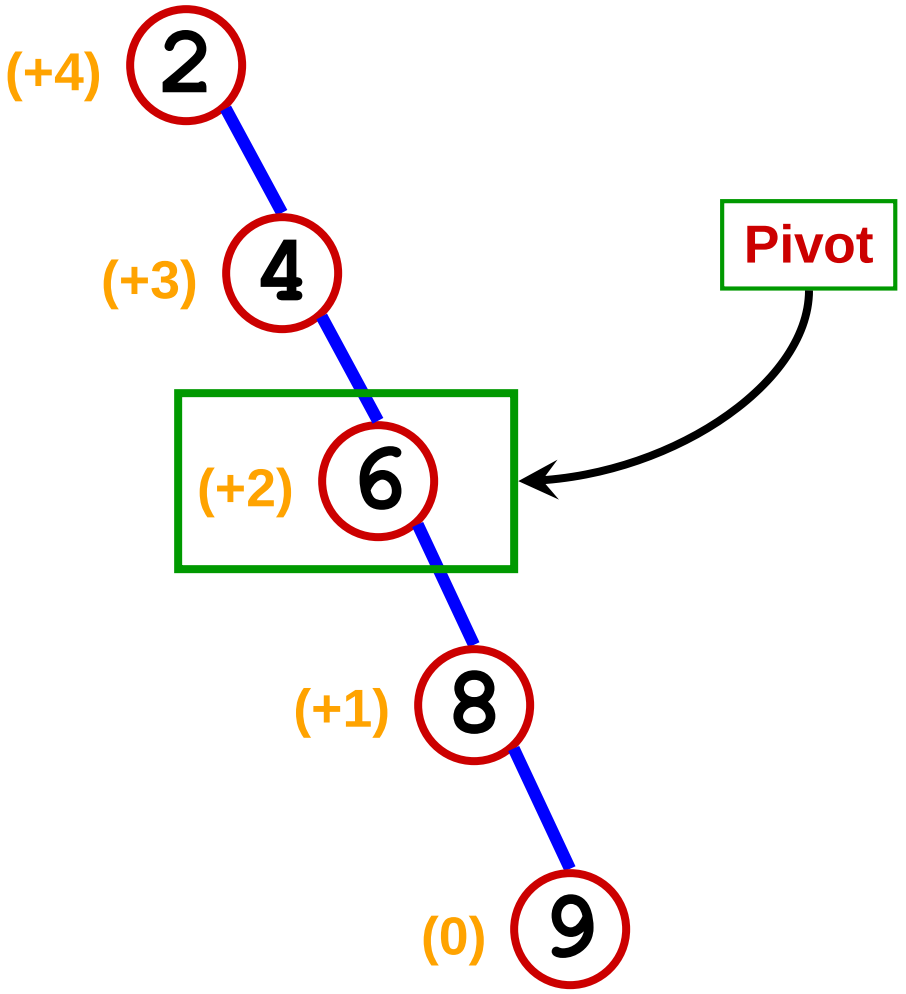
AVL Trees Example – 1

UNIVERSITY OF CALGARY

First try to do it by yourself

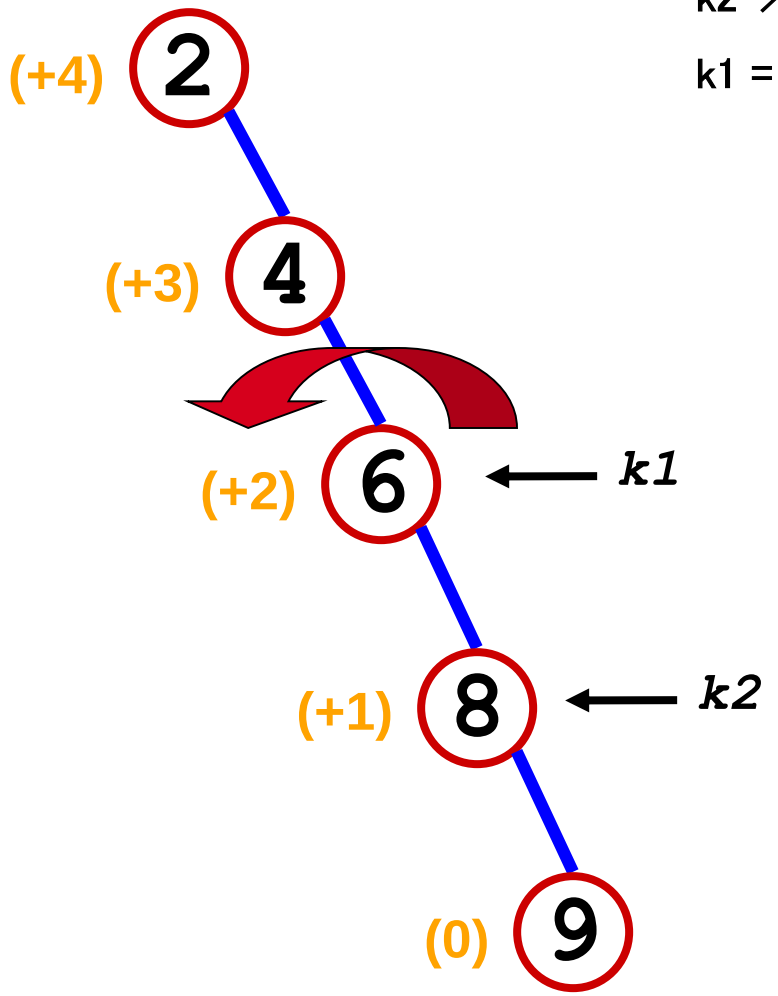



(+4)
2
(+3)
4
(+2)
6
(+1)
8
(0)
9


(+4)
2
(+3)
4
Pivot
(+2)
6
(+1)
8
(0)
9

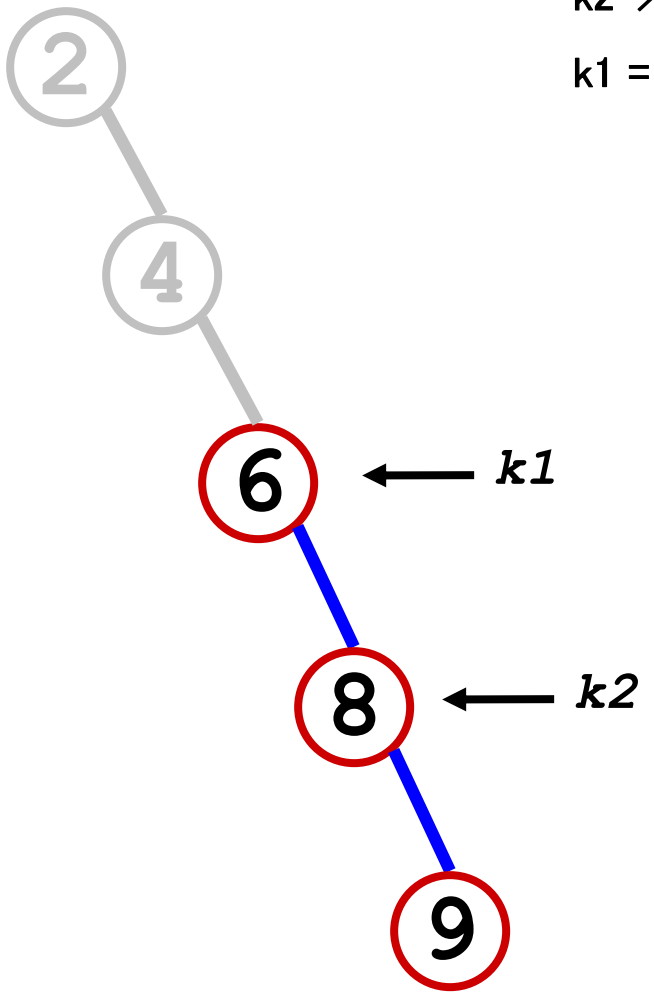
Single Left Rotation

```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```



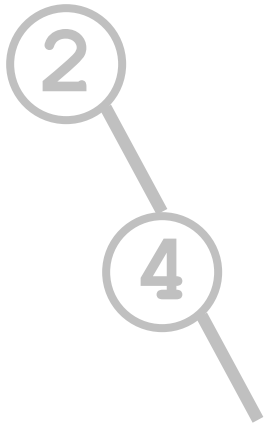
Single Left Rotation

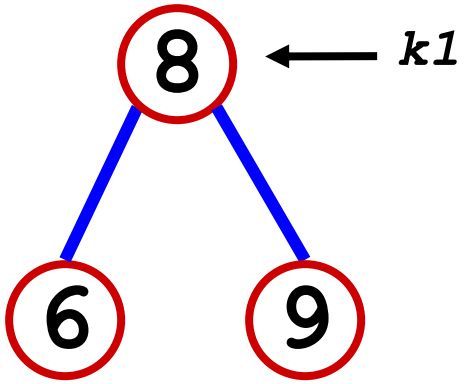
```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```

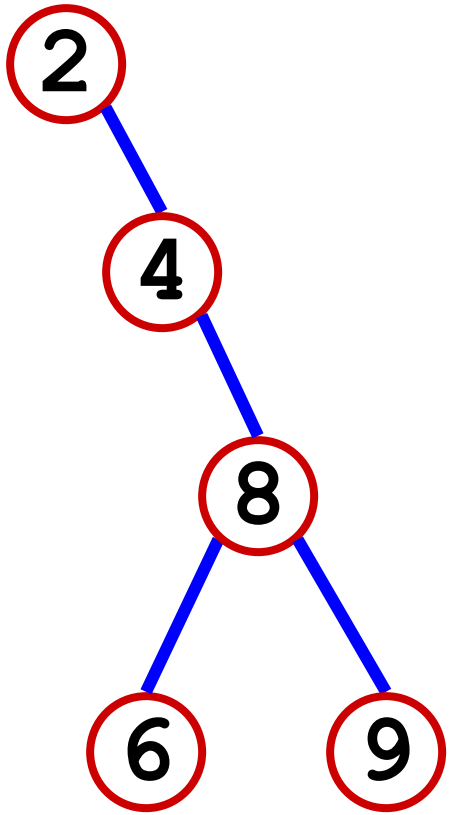


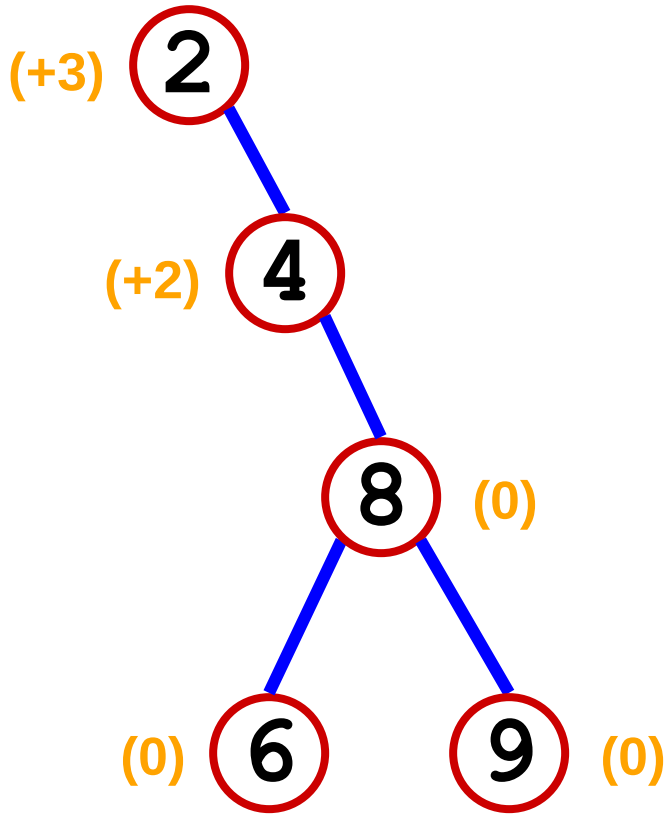
Single Left Rotation

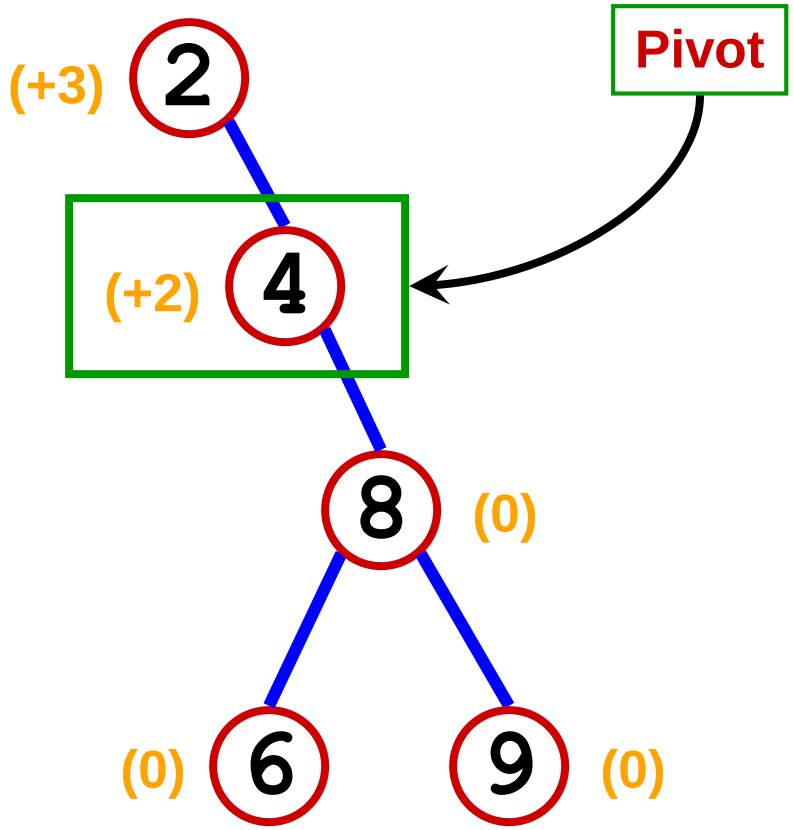
```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```



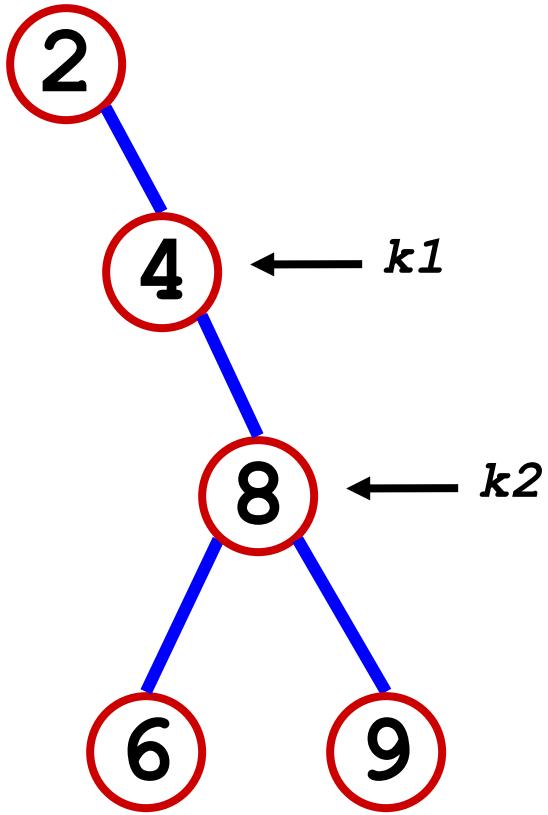



2
4
8
6
9


(+3)
2
(+2)
4
8
(0)
(0)
6
9
(0)

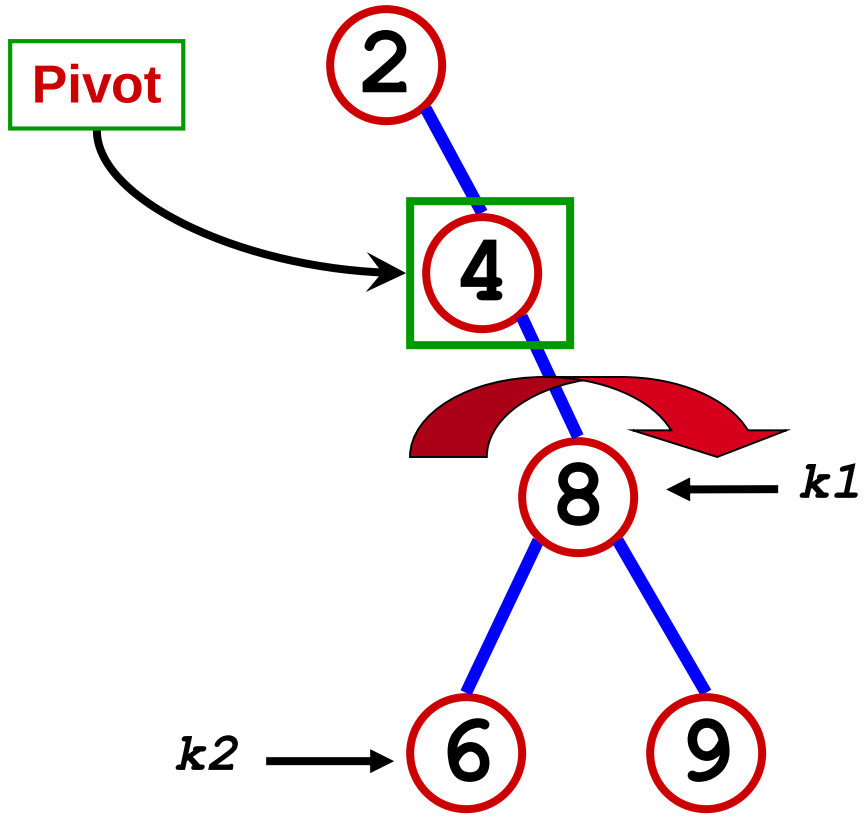

Pivot
(+3)
2
(+2)
4
8
(0)
(0)
6
9
(0)

Double Rotation



Double Rotation

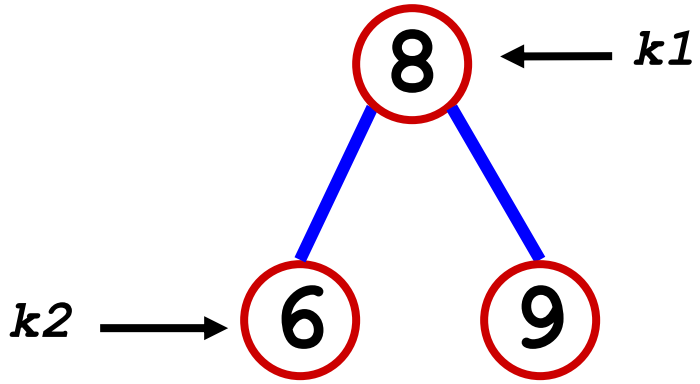
```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```



Step 1: Rotate child and grandchild of pivot

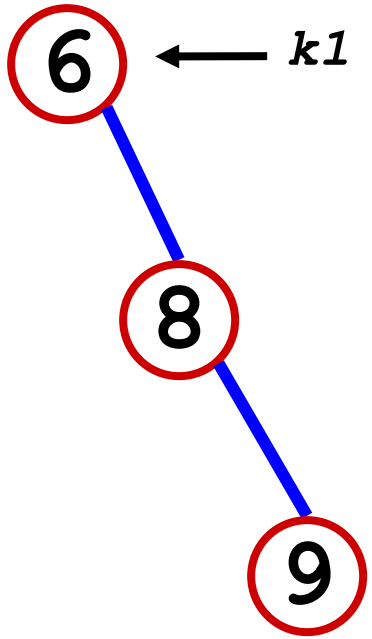
Double Rotation

```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```



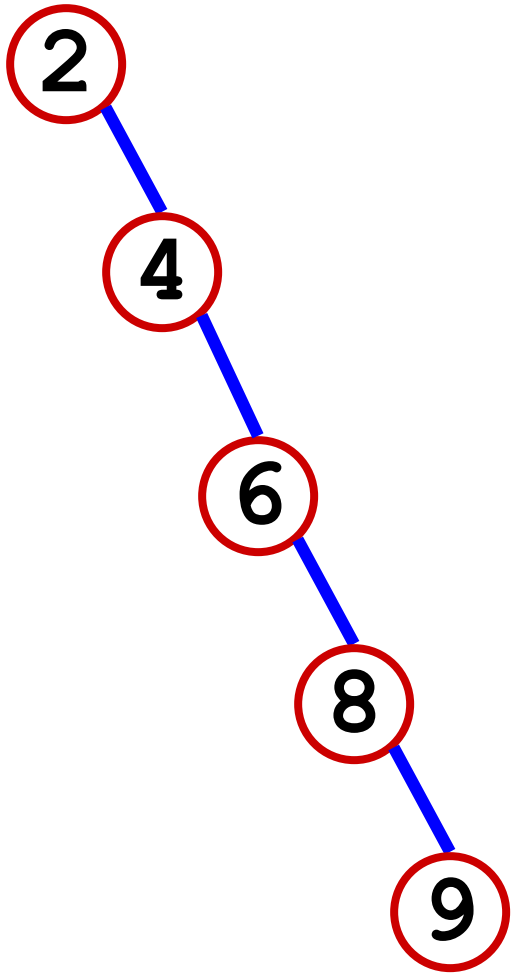
Double Rotation

```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```



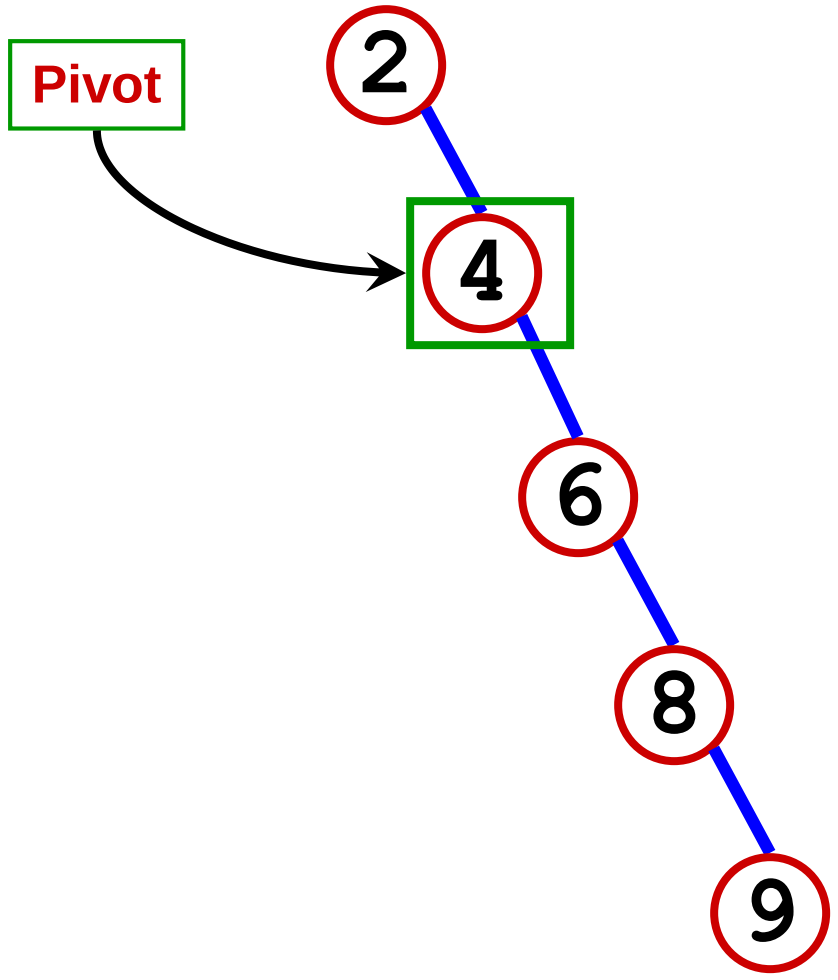
Double Rotation

Step 2: Rotate node and new child (AVL)



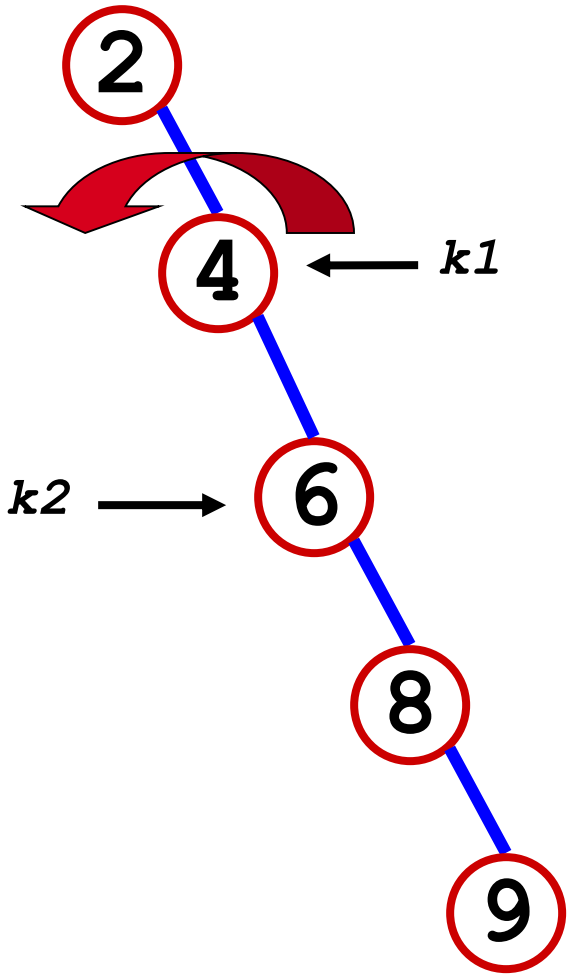
Double Rotation

Step 2: Rotate node and new child (AVL)



Double Rotation

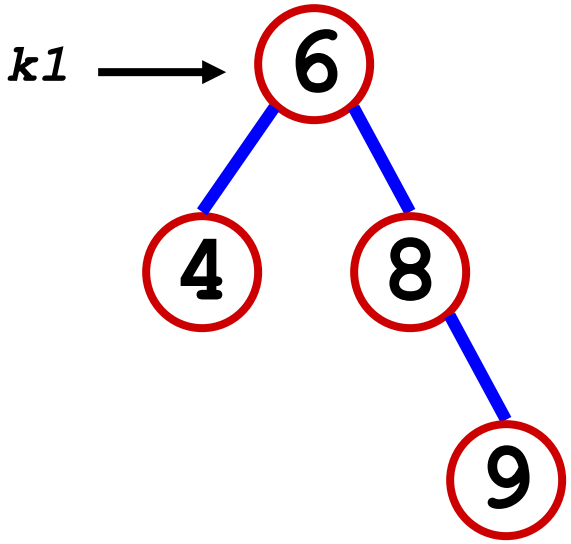
```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```

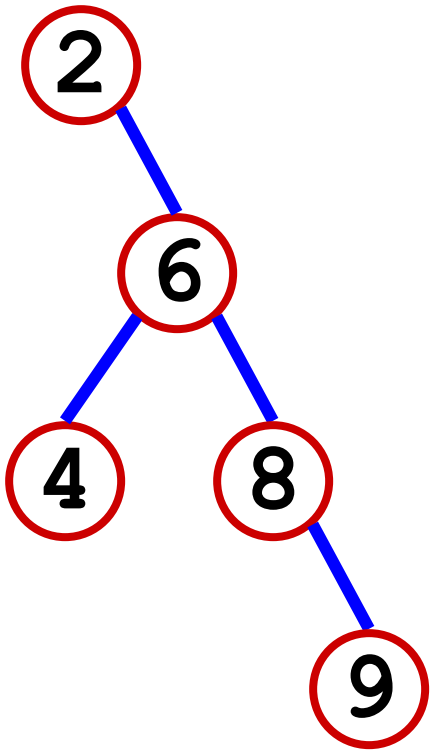


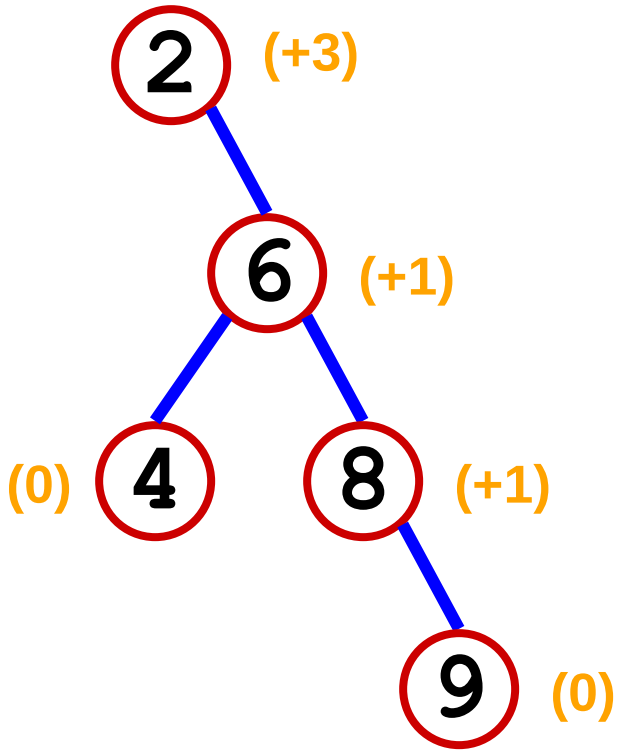
Step 2: Rotate node and new child (AVL)

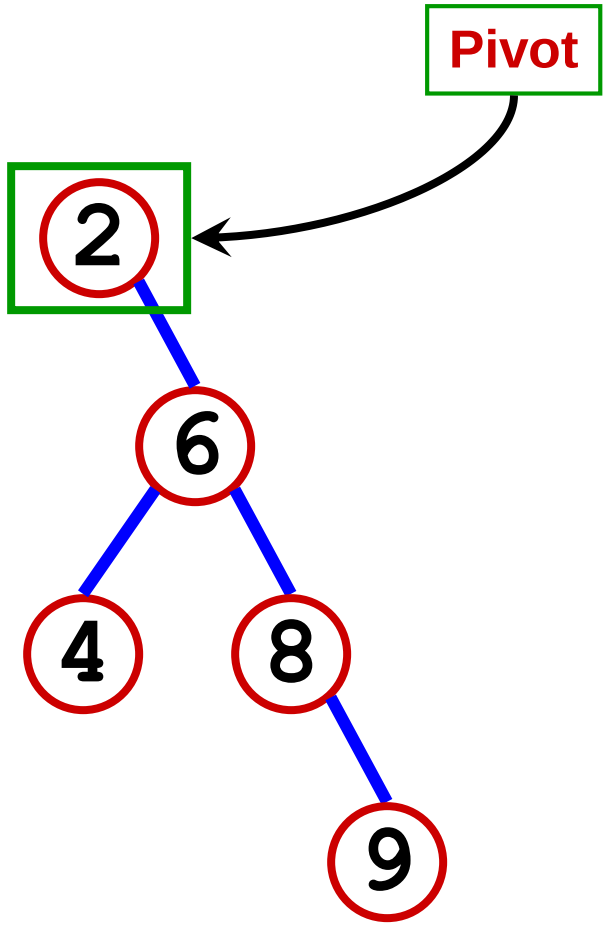
Double Rotation

```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```



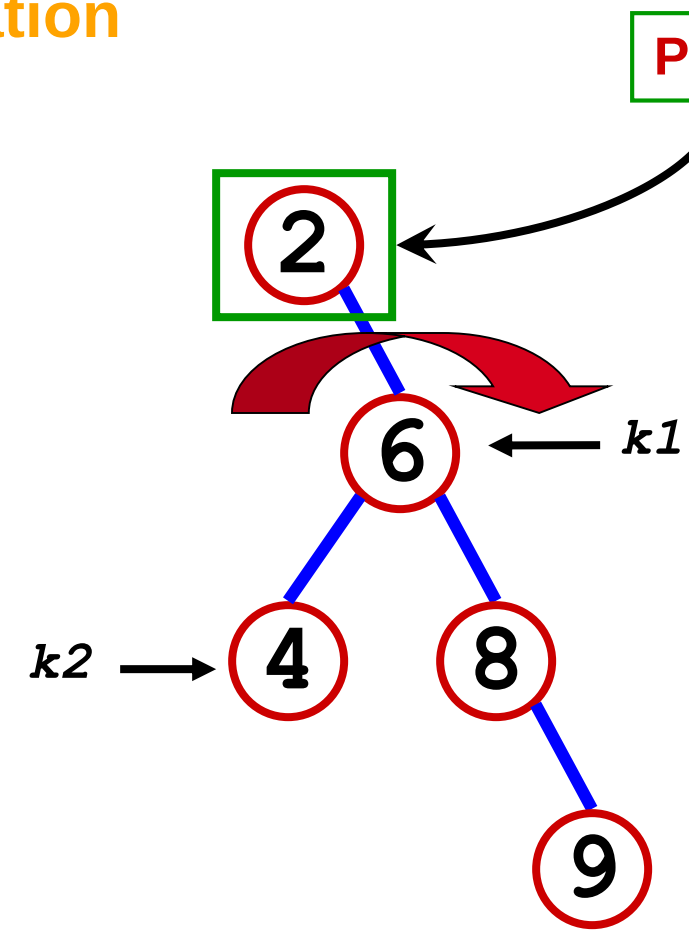

2
6
4
8
9


2
(+3)
6
(+1)
(0)
4
8
(+1)
9
(0)



UNIVERSITY OF CALGARY

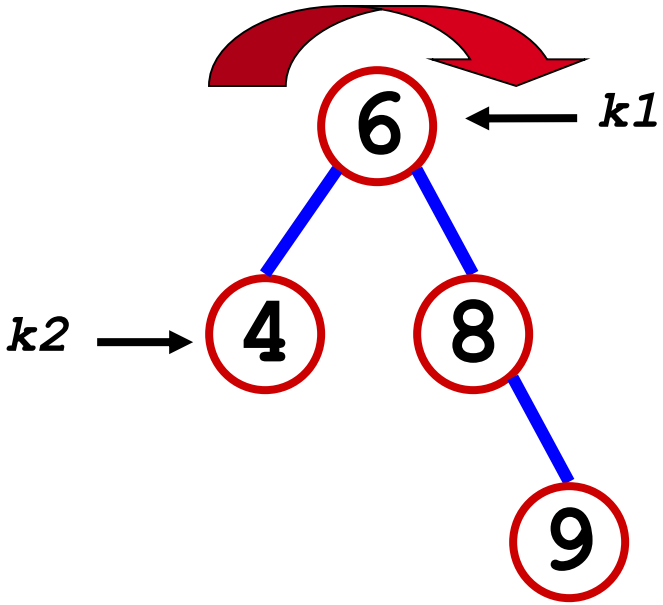
Double Rotation



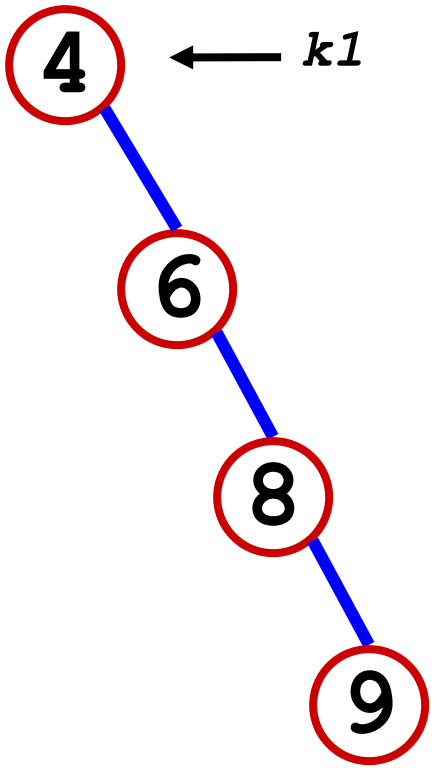

```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```

Double Rotation

```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```

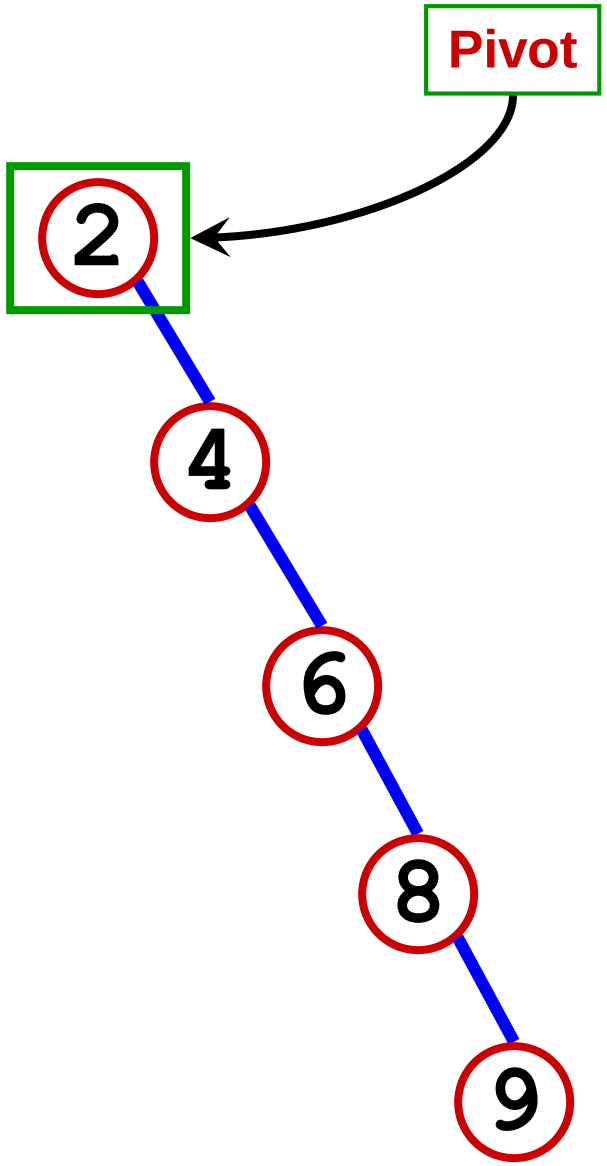


Double Rotation

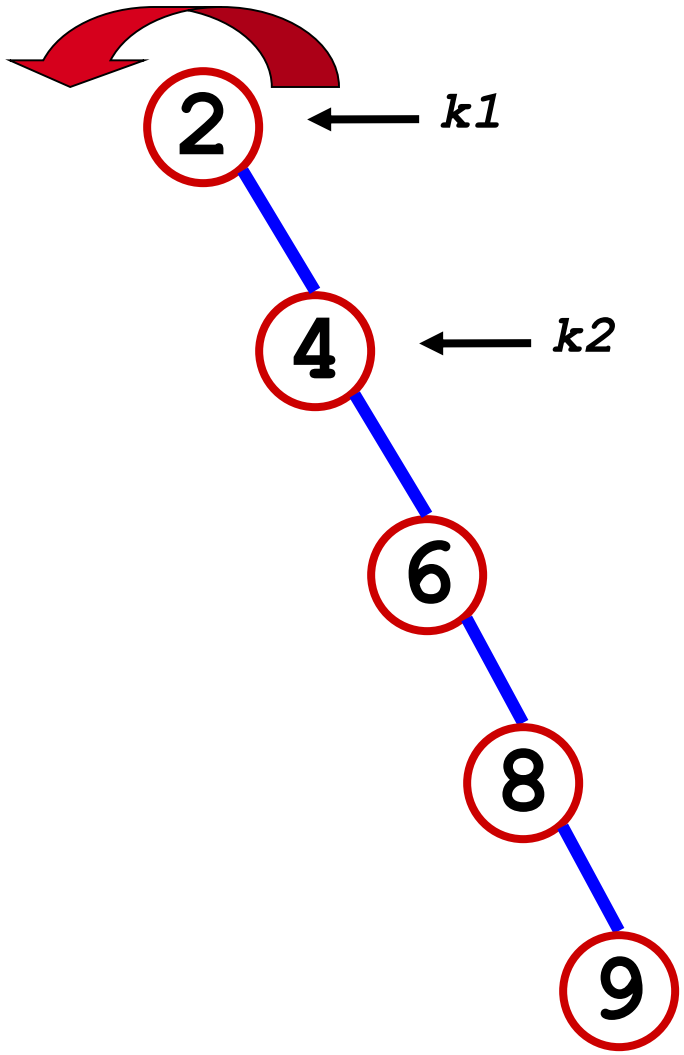



```
BinaryNode *k2 = k1->left;
k1->left = k2->right;
k2->right = k1;
k1 = k2;
```

Double Rotation



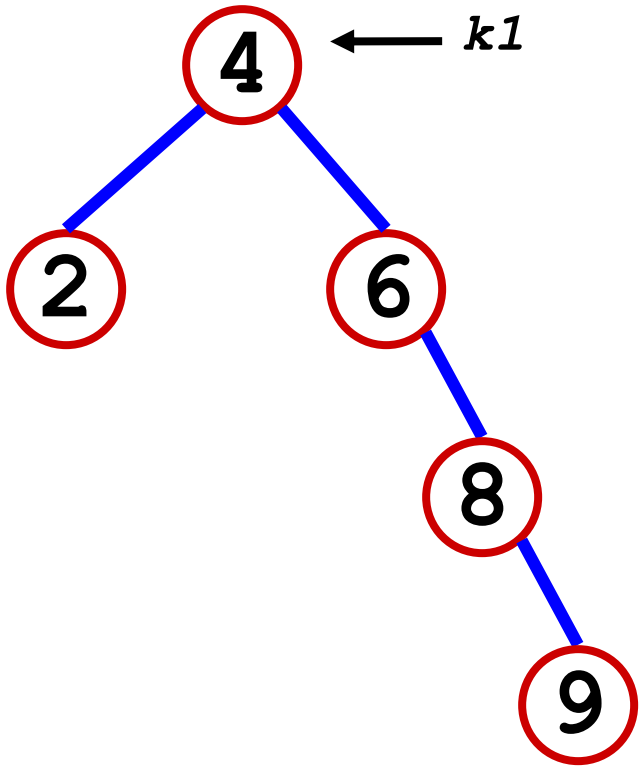
Double Rotation

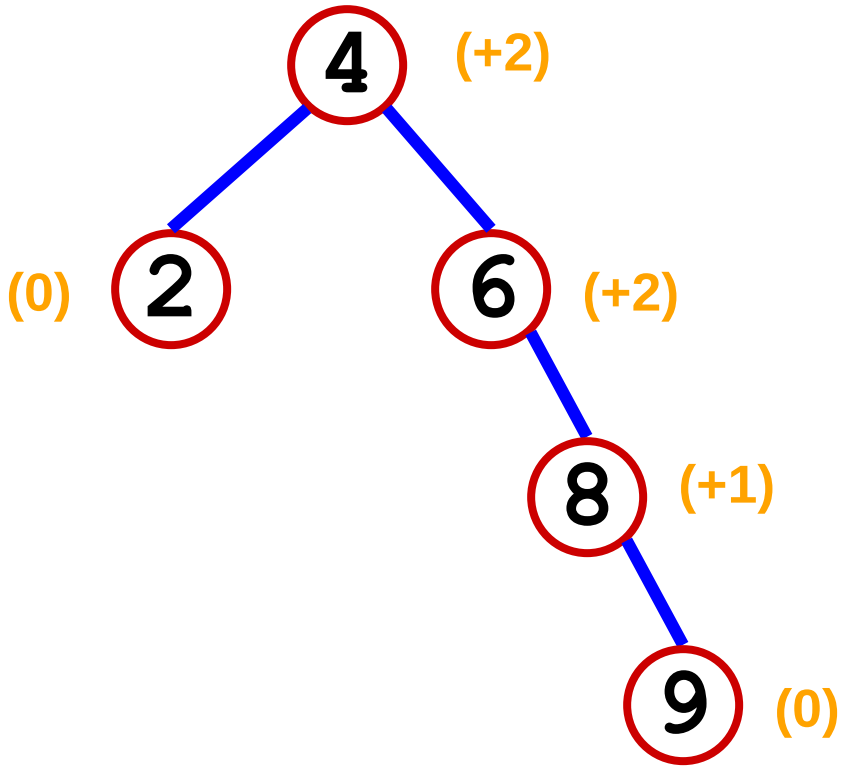



```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```

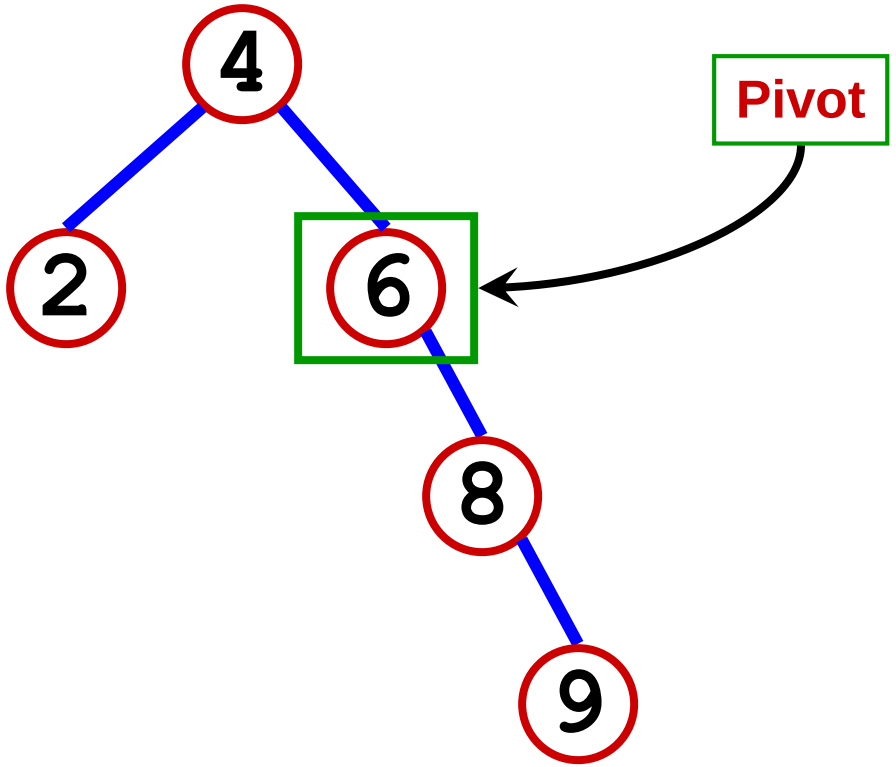
Double Rotation

```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```



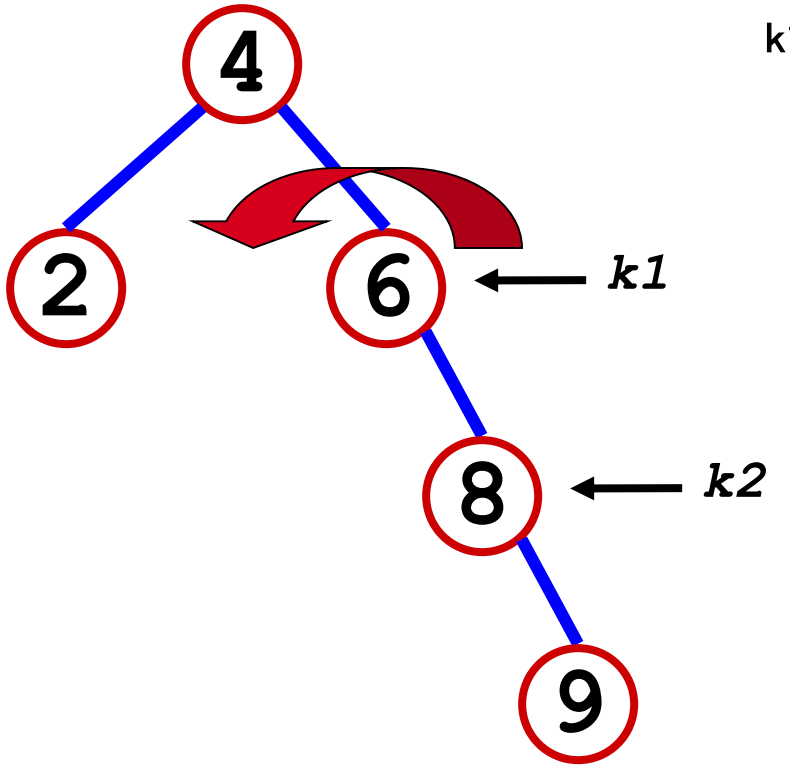

4
(+2)
(0)
2
6
(+2)
8
(+1)
9
(0)

Single Left Rotation



Single Left Rotation

```
BinaryNode *k2 = k1->right;
k1->right = k2->left;
k2->left = k1;
k1 = k2;
```



DONE!