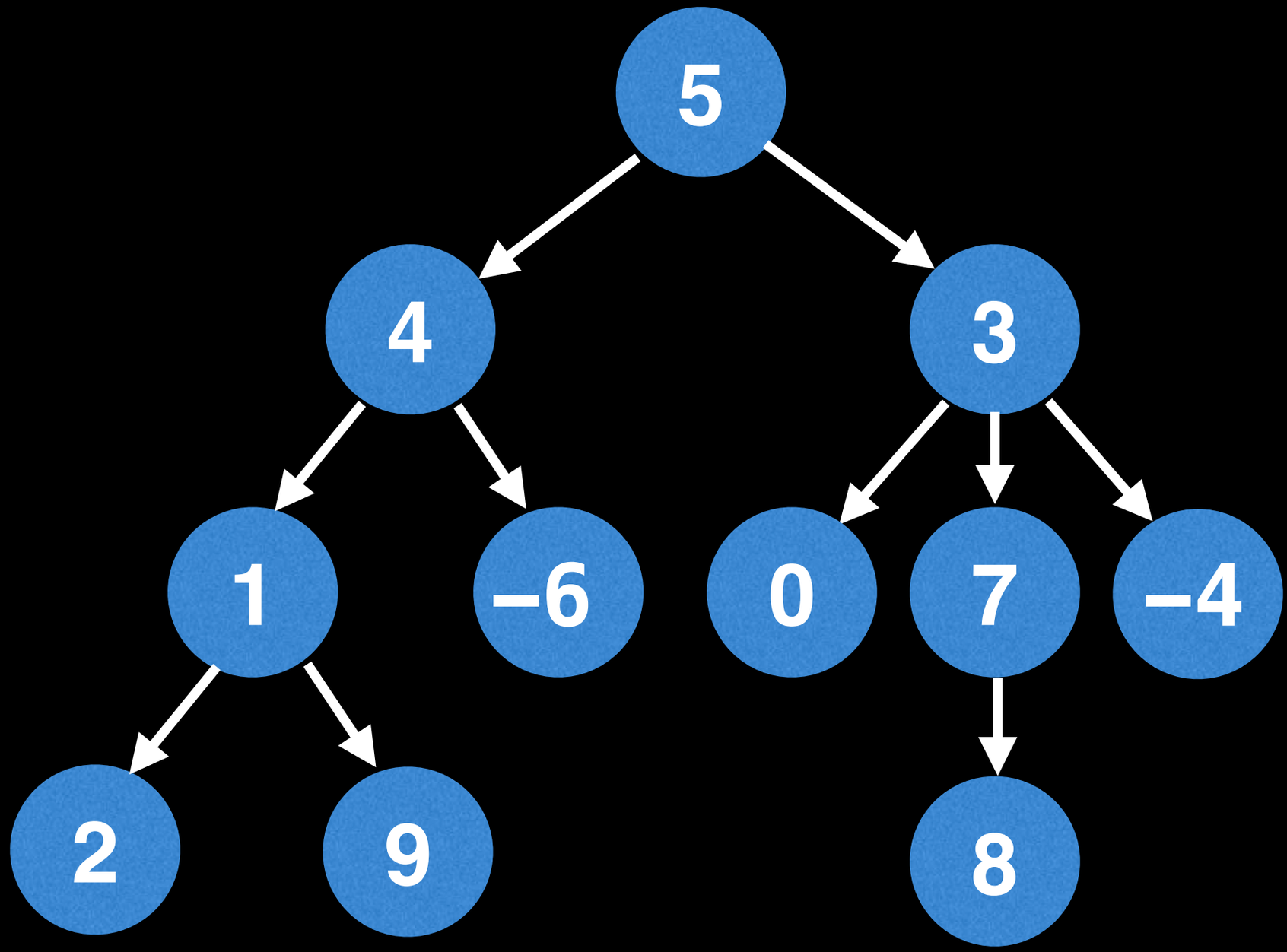
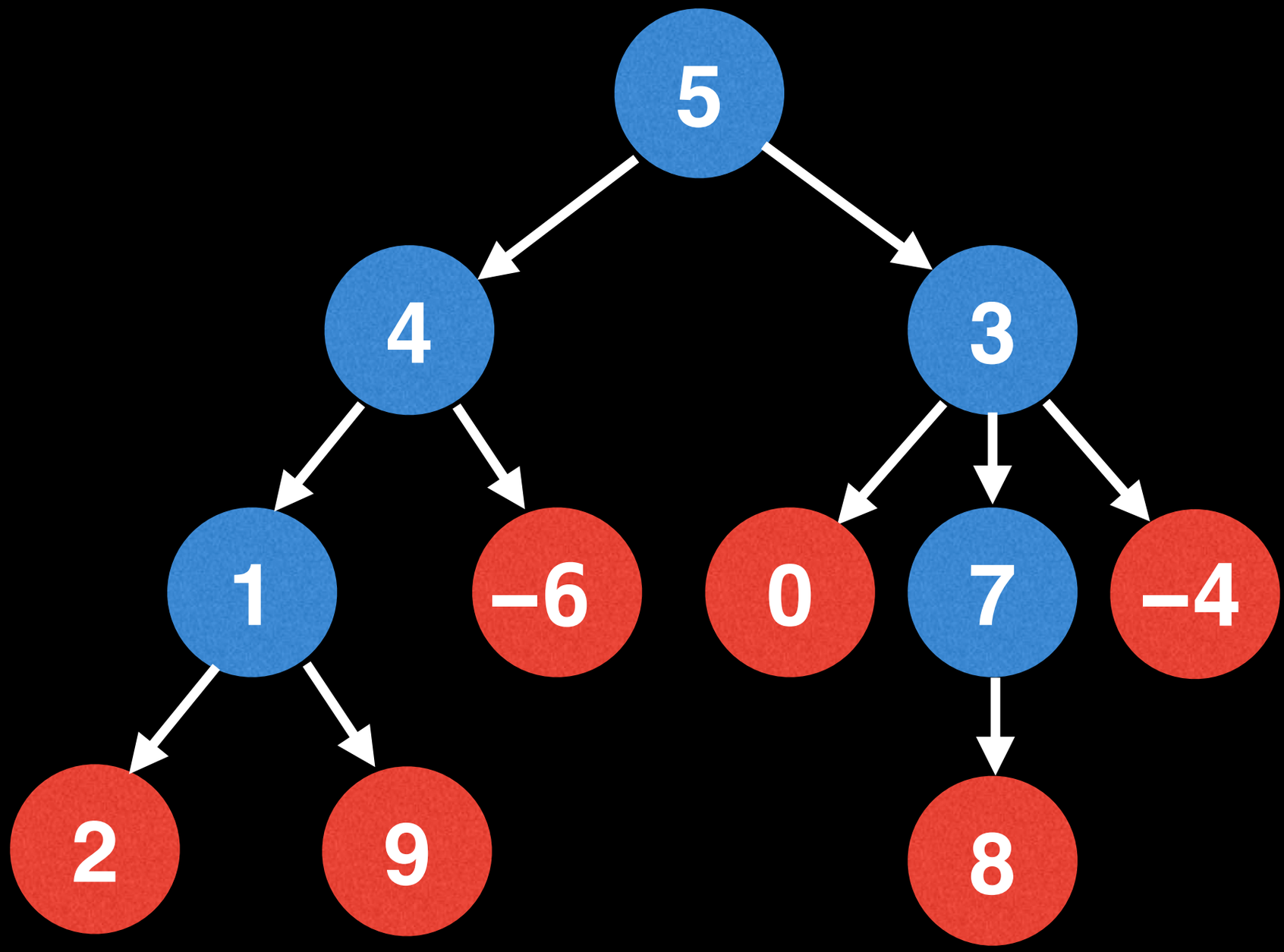
Beginner tree algorithms

William Fiset

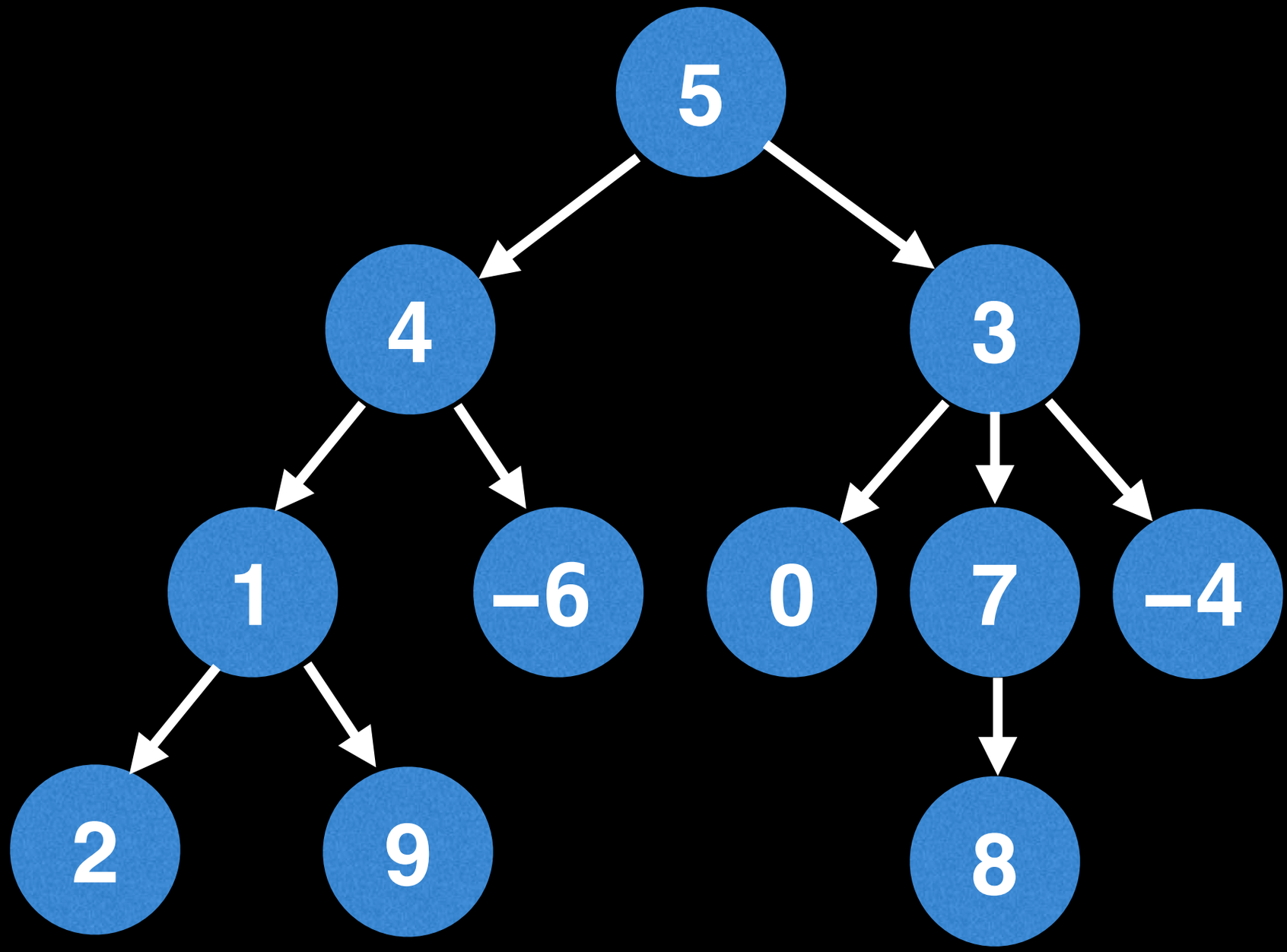
Problem 1: leaf node sum

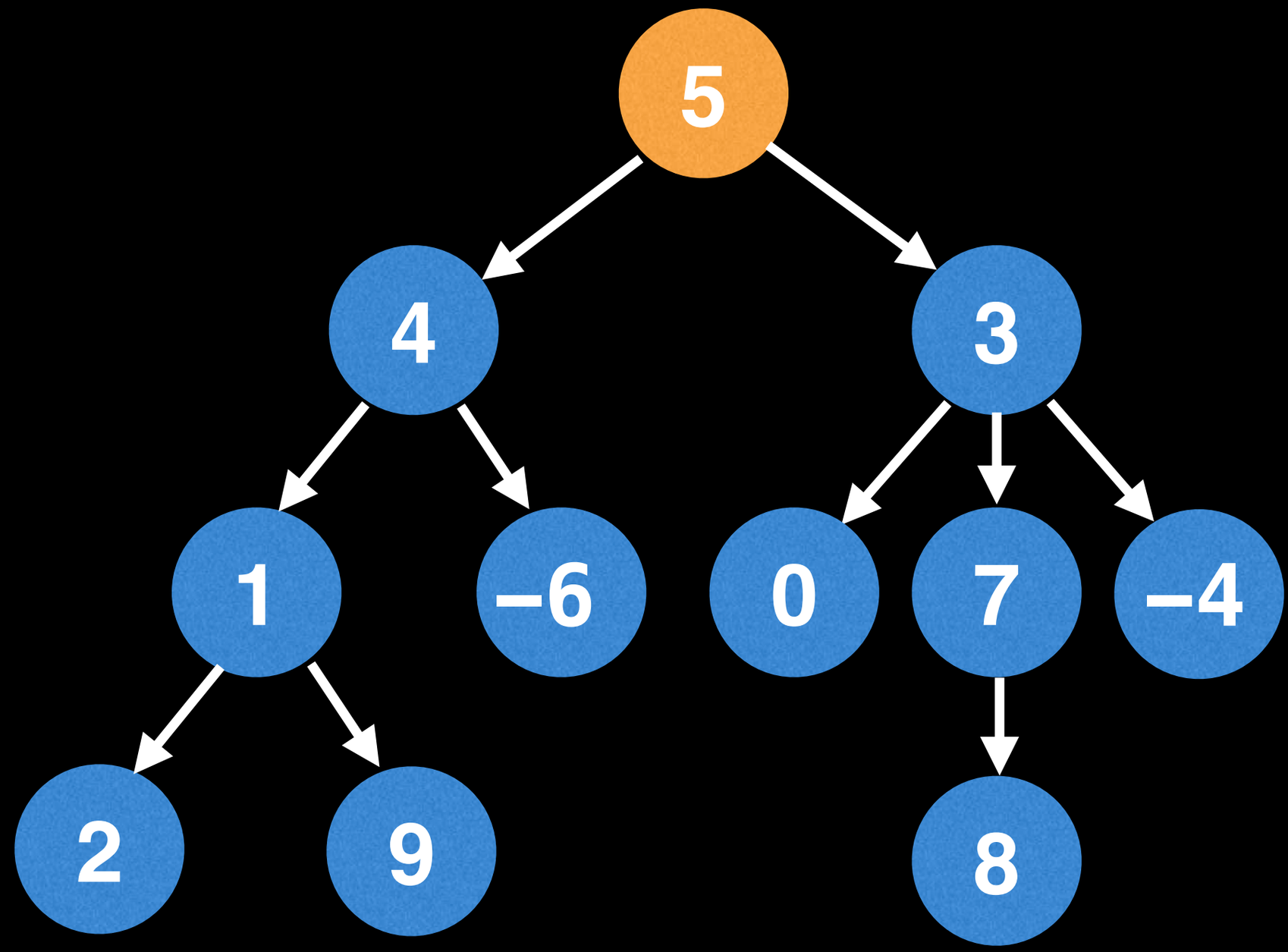
What is the sum of all the leaf node values in a tree?


5
4
3
1
−6
0
7
−4
2
9
8

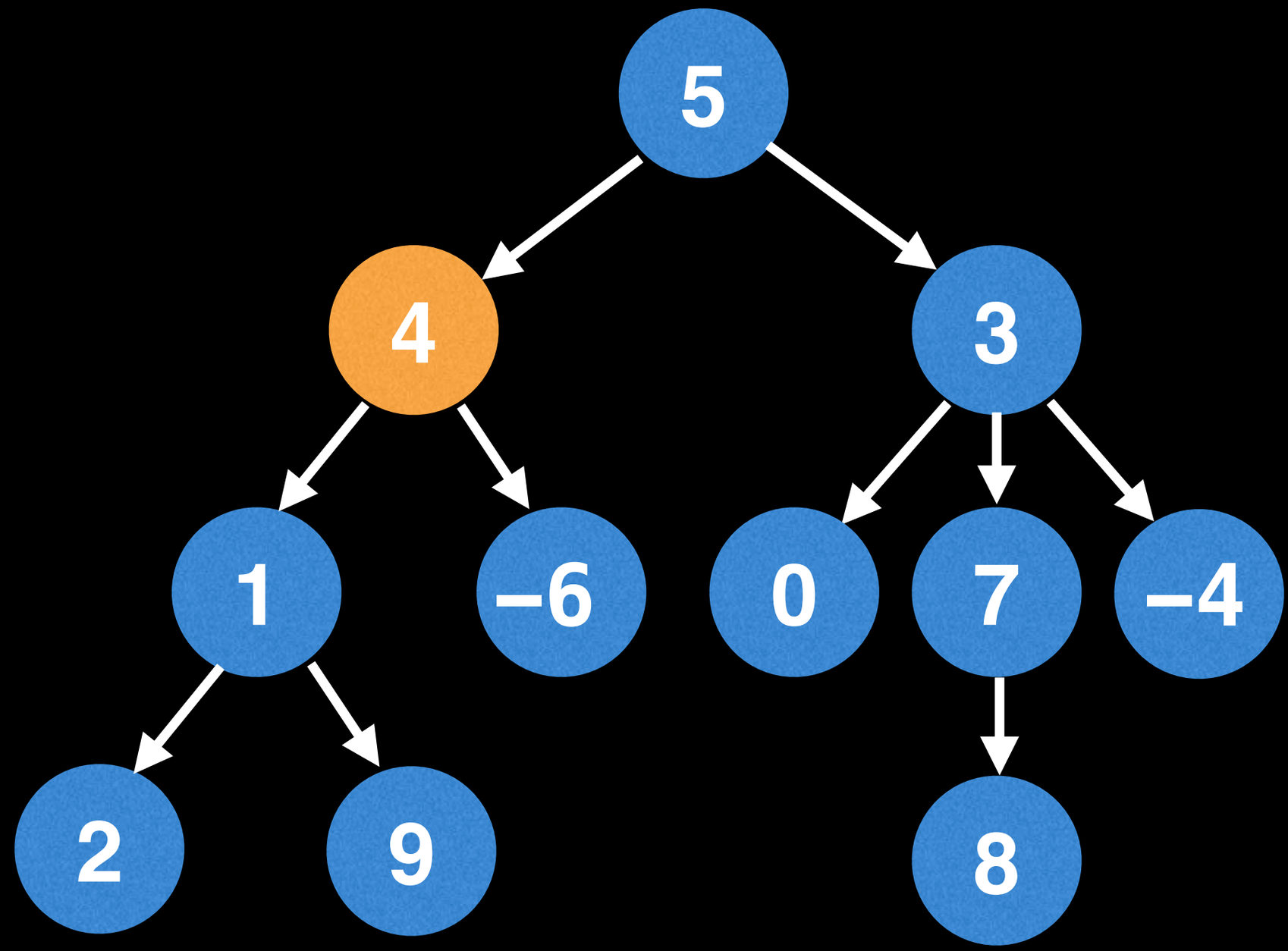


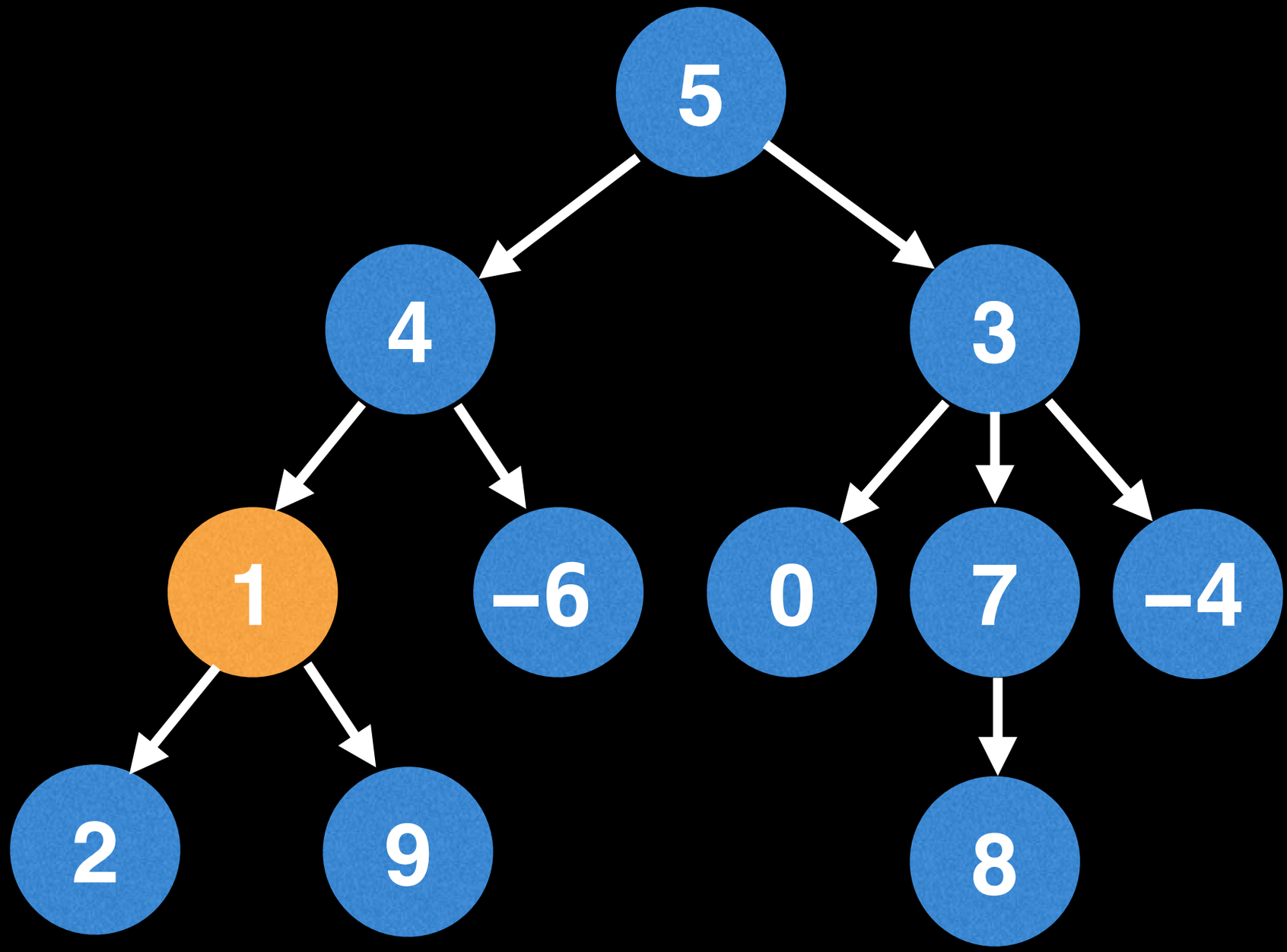
$$2 + 9 - 6 + 0 + 8 - 4 = 9$$

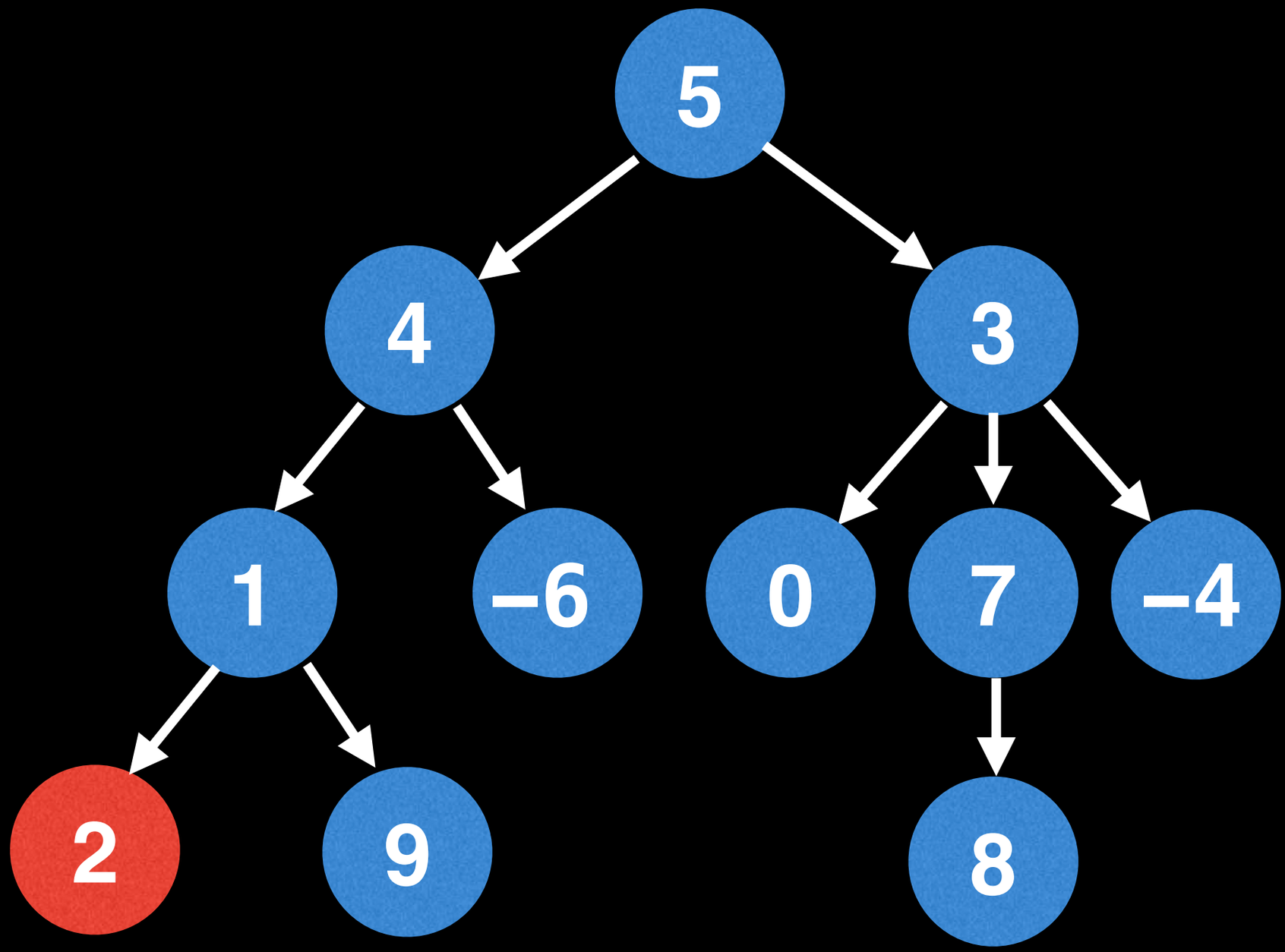

5
4
3
1
−6
0
7
−4
2
9
8

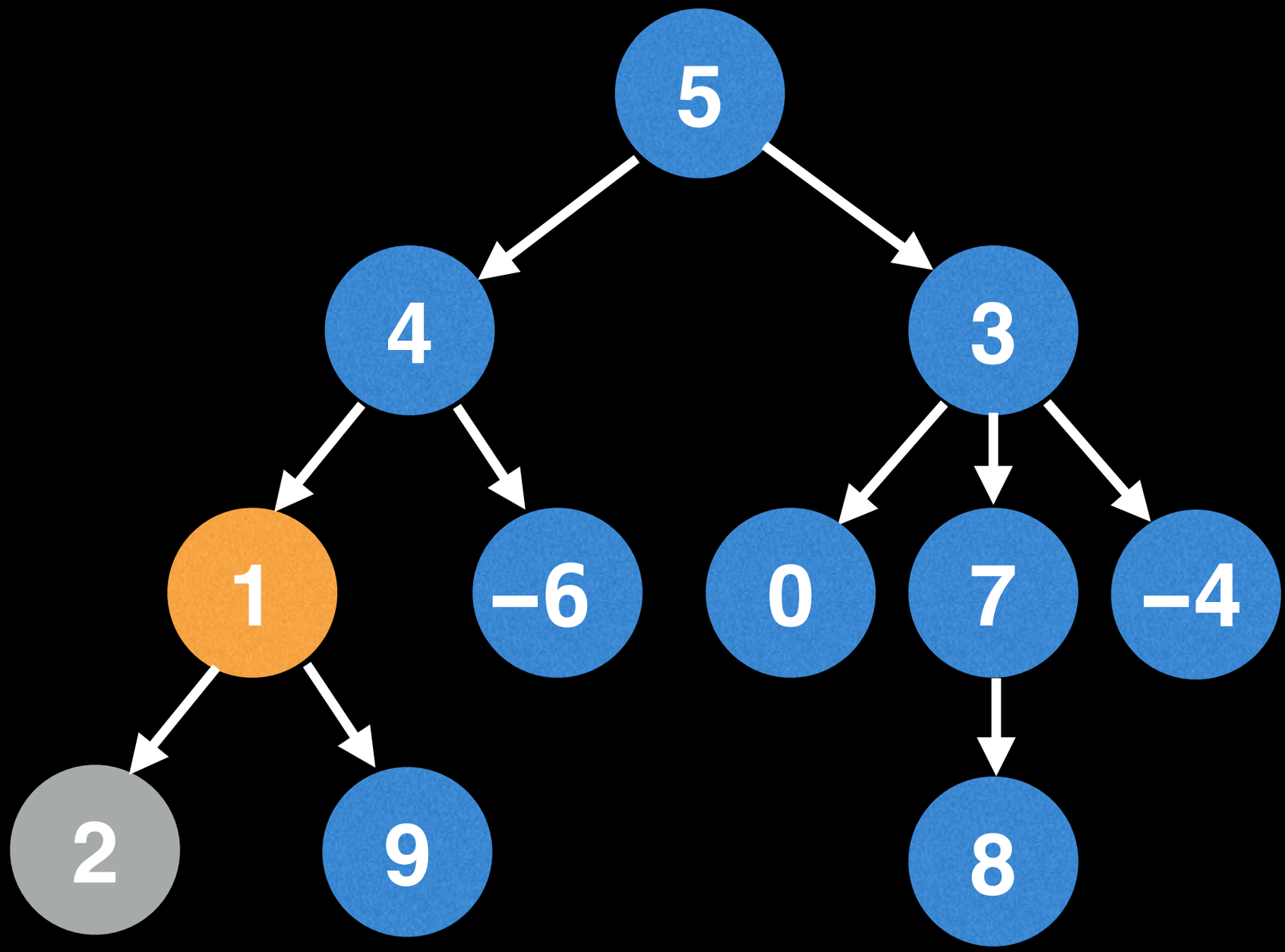


When dealing with rooted trees you begin with having a reference to the root node as a starting point for most algorithms.

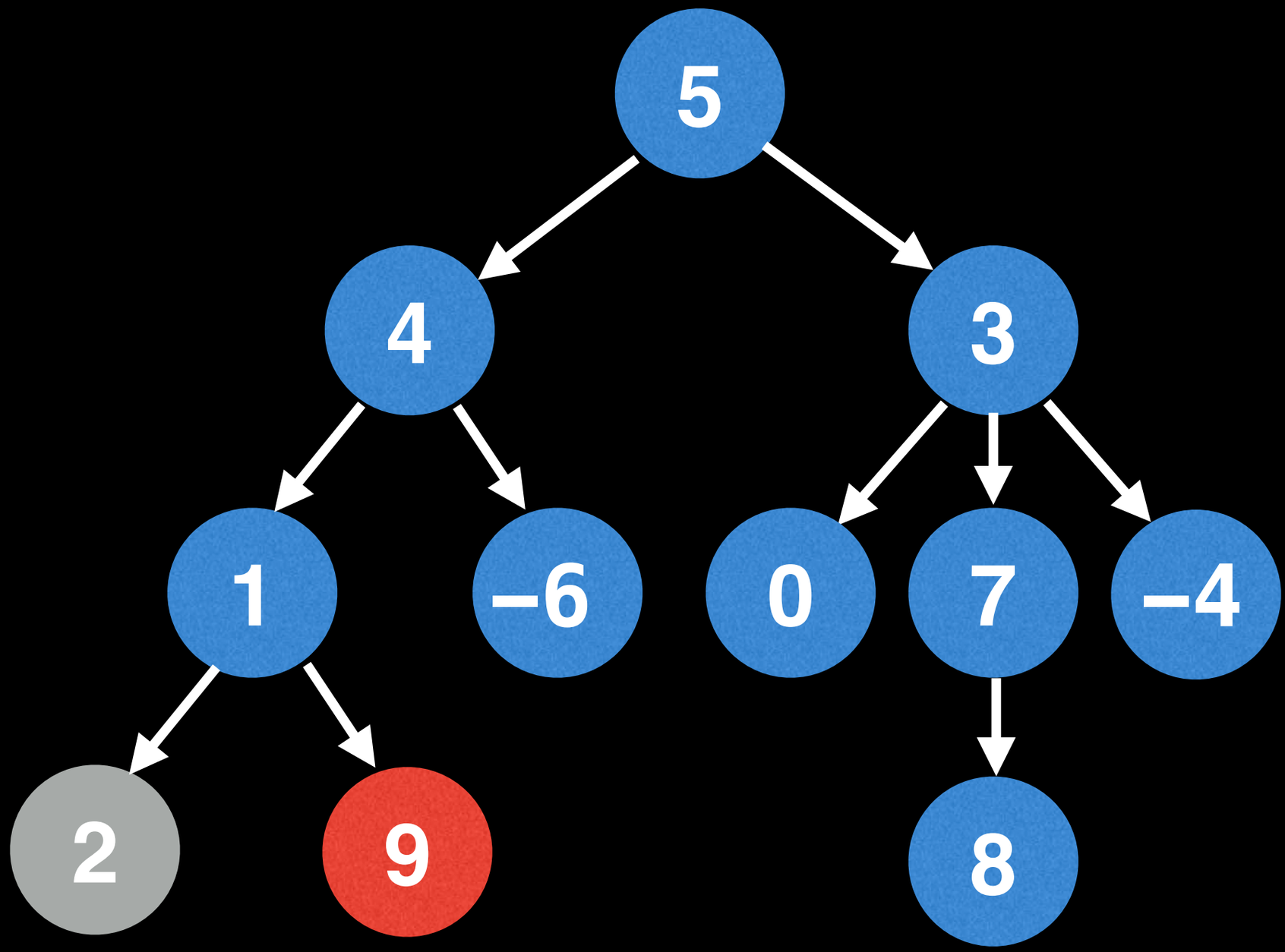

5
4
3
1
−6
0
7
−4
2
9
8


5
4
3
1
−6
0
7
−4
2
9
8

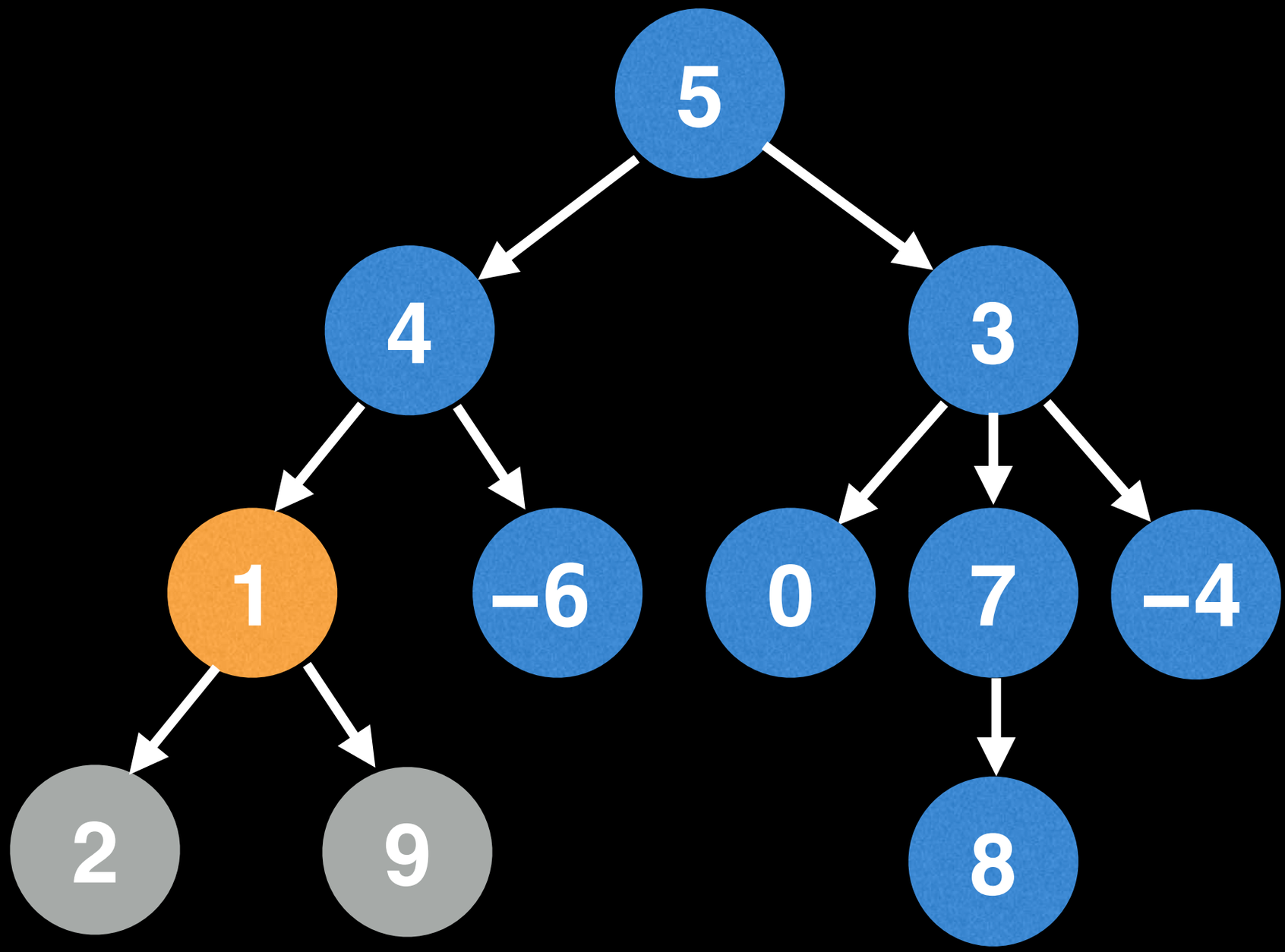

5
4
3
1
−6
0
7
−4
2
9
8



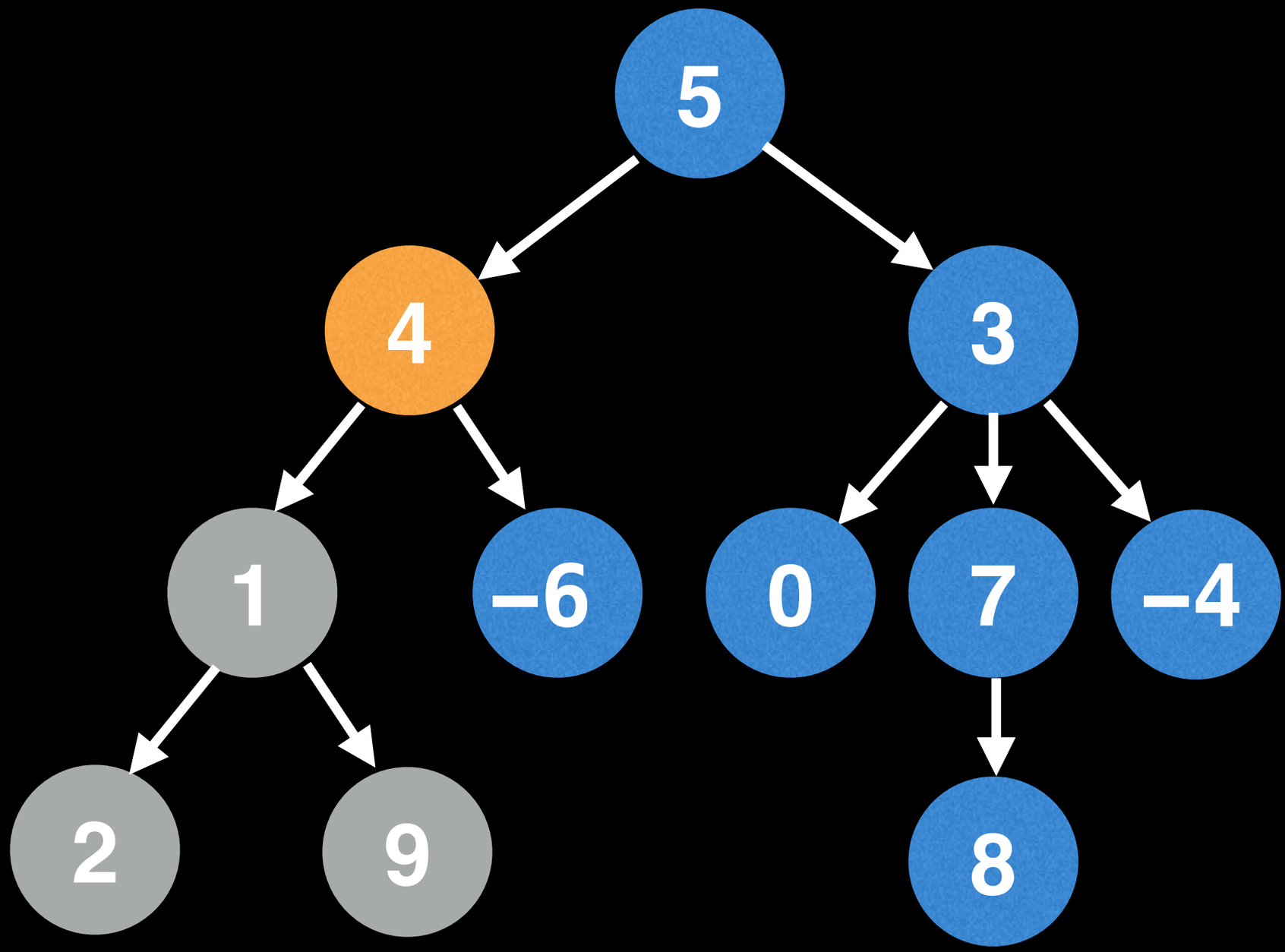
2



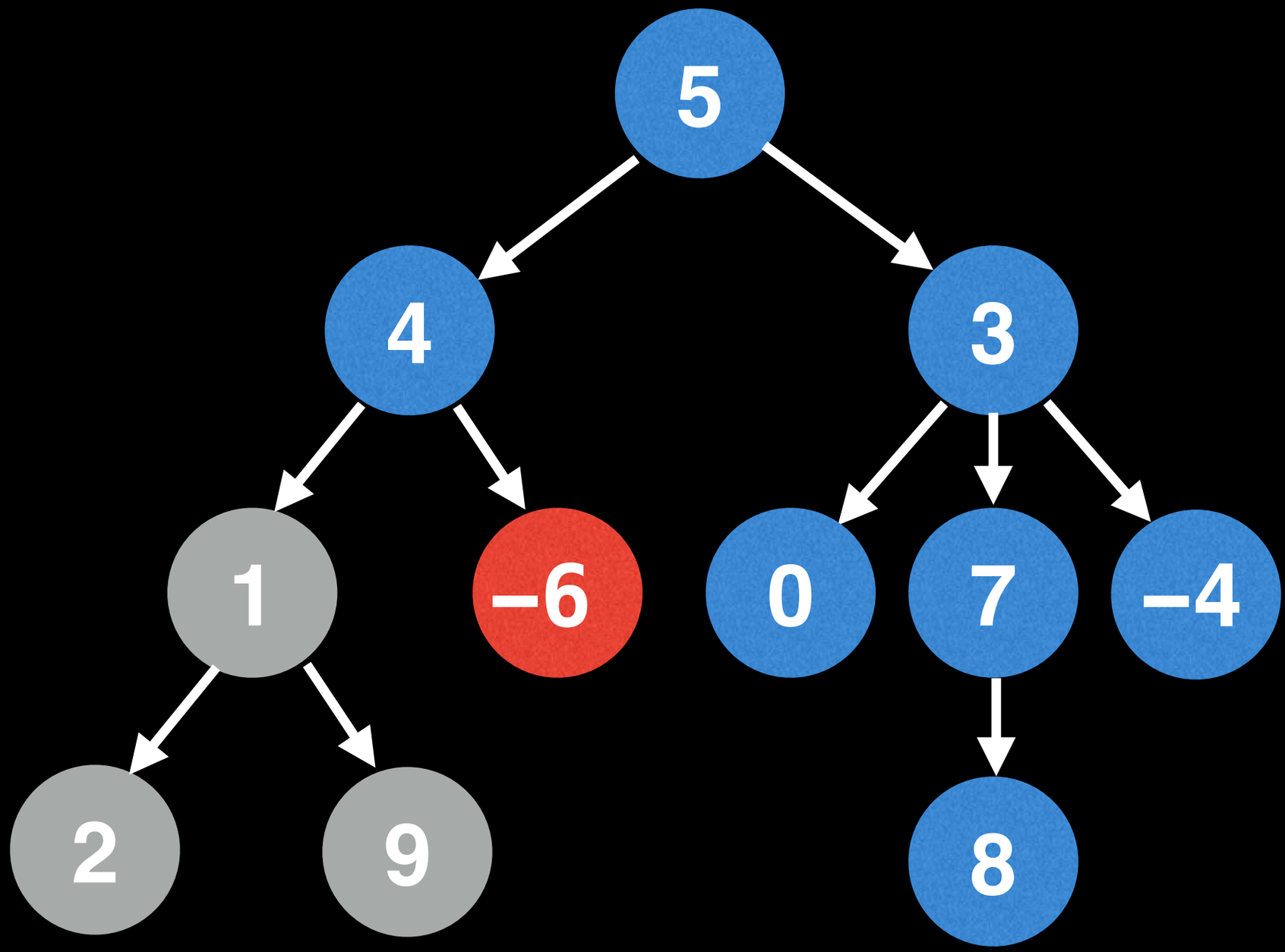
2



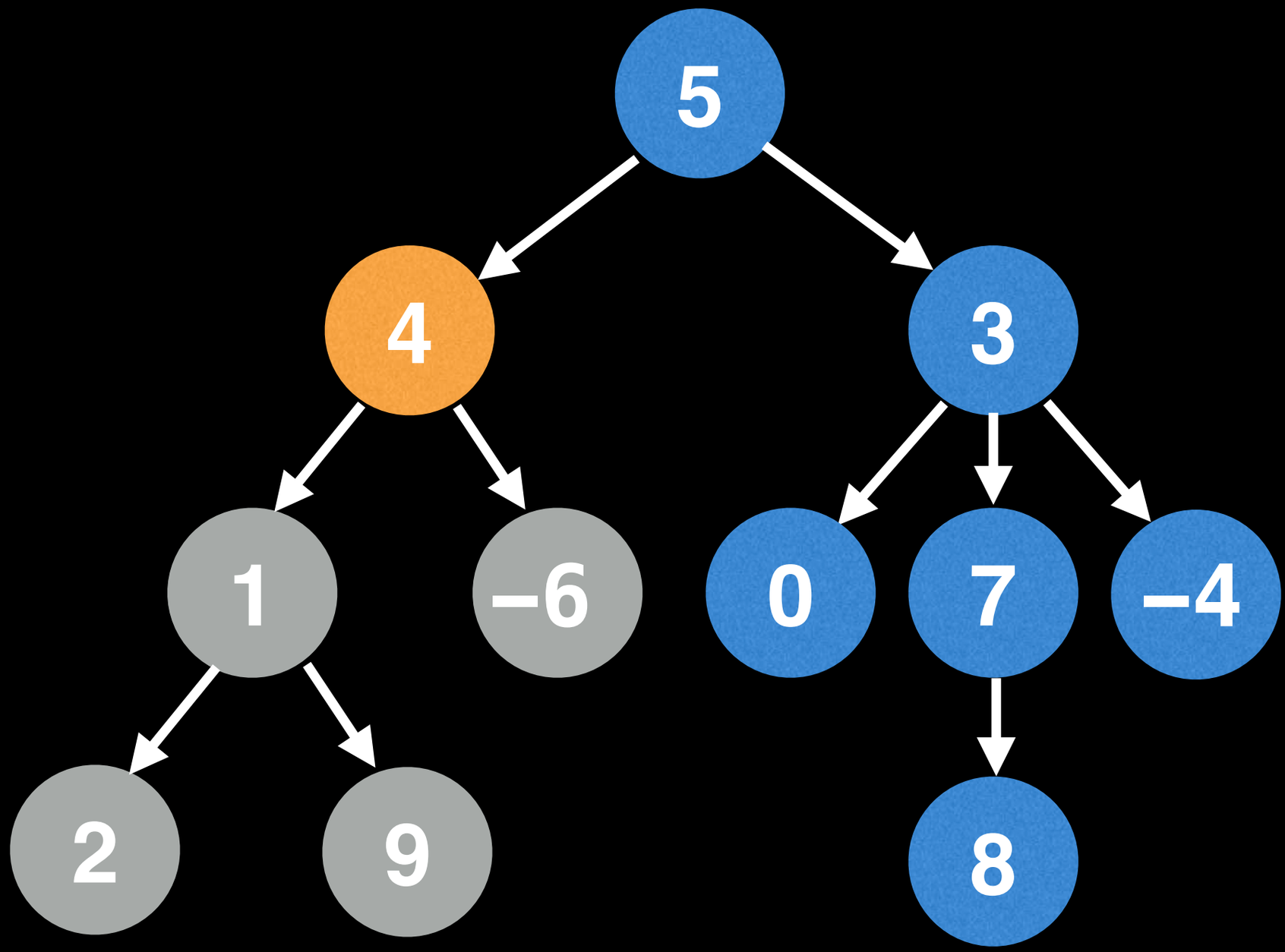
2 + 9



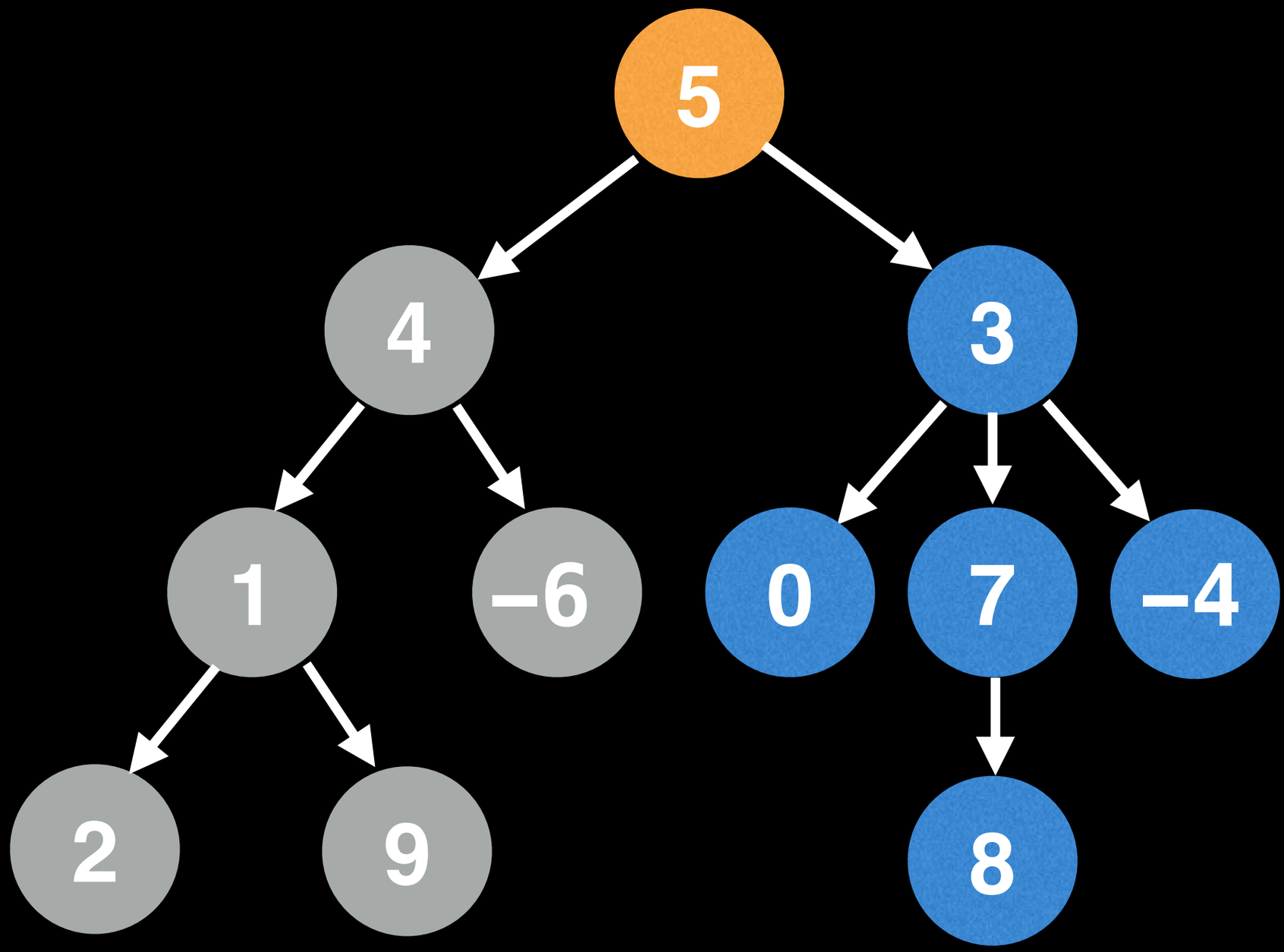
2 + 9



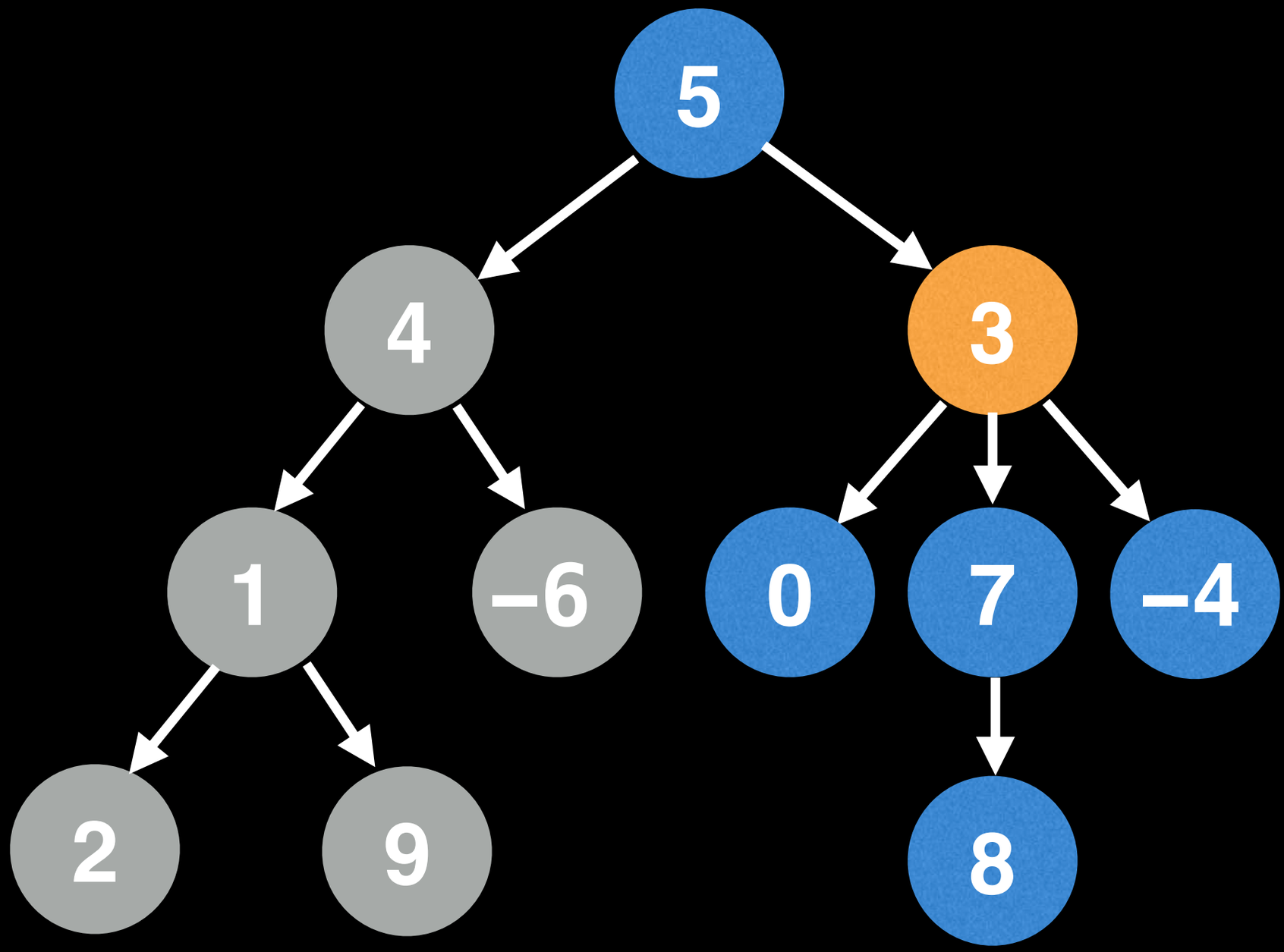
2 + 9



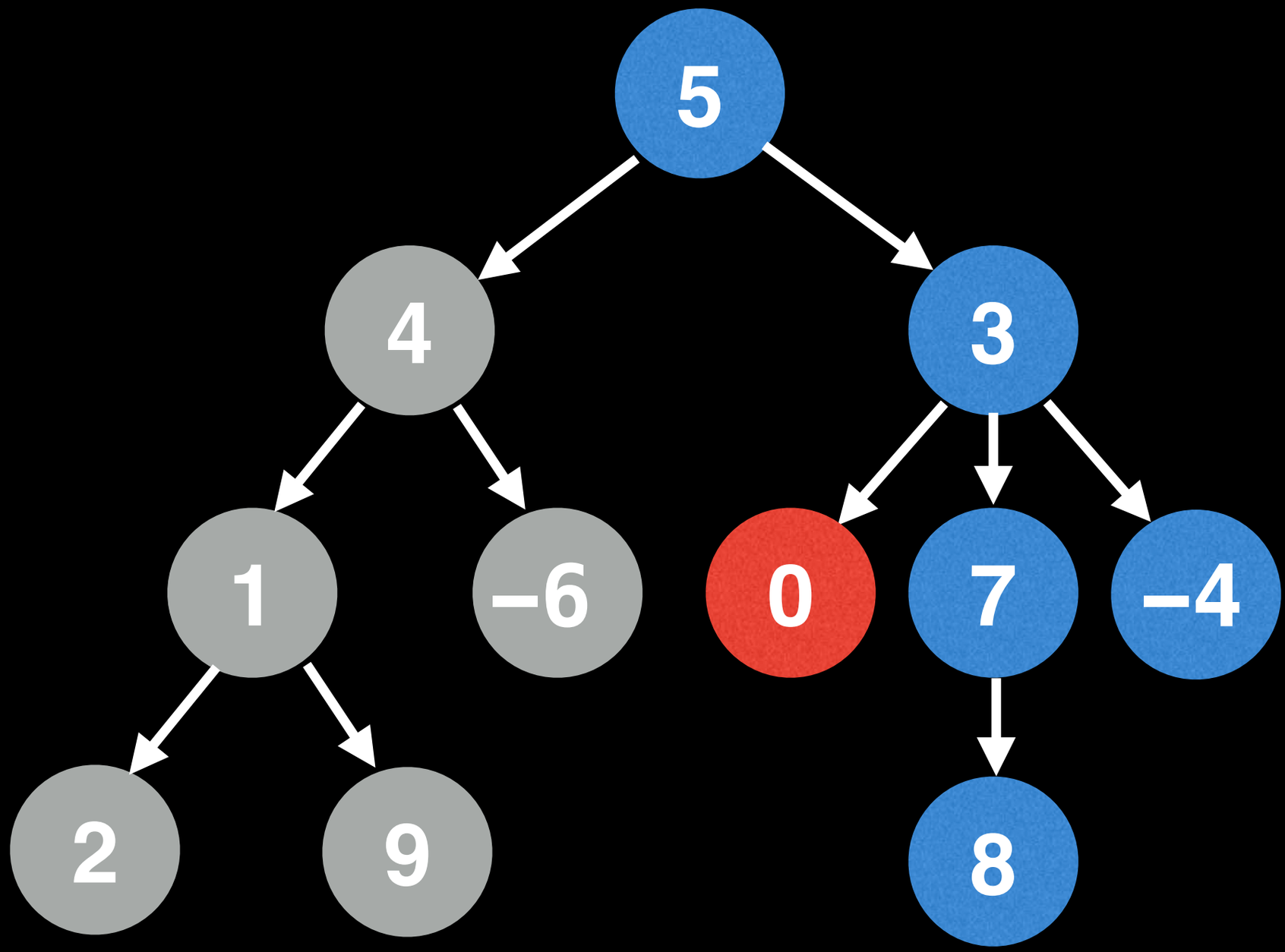
2 + 9 − 6



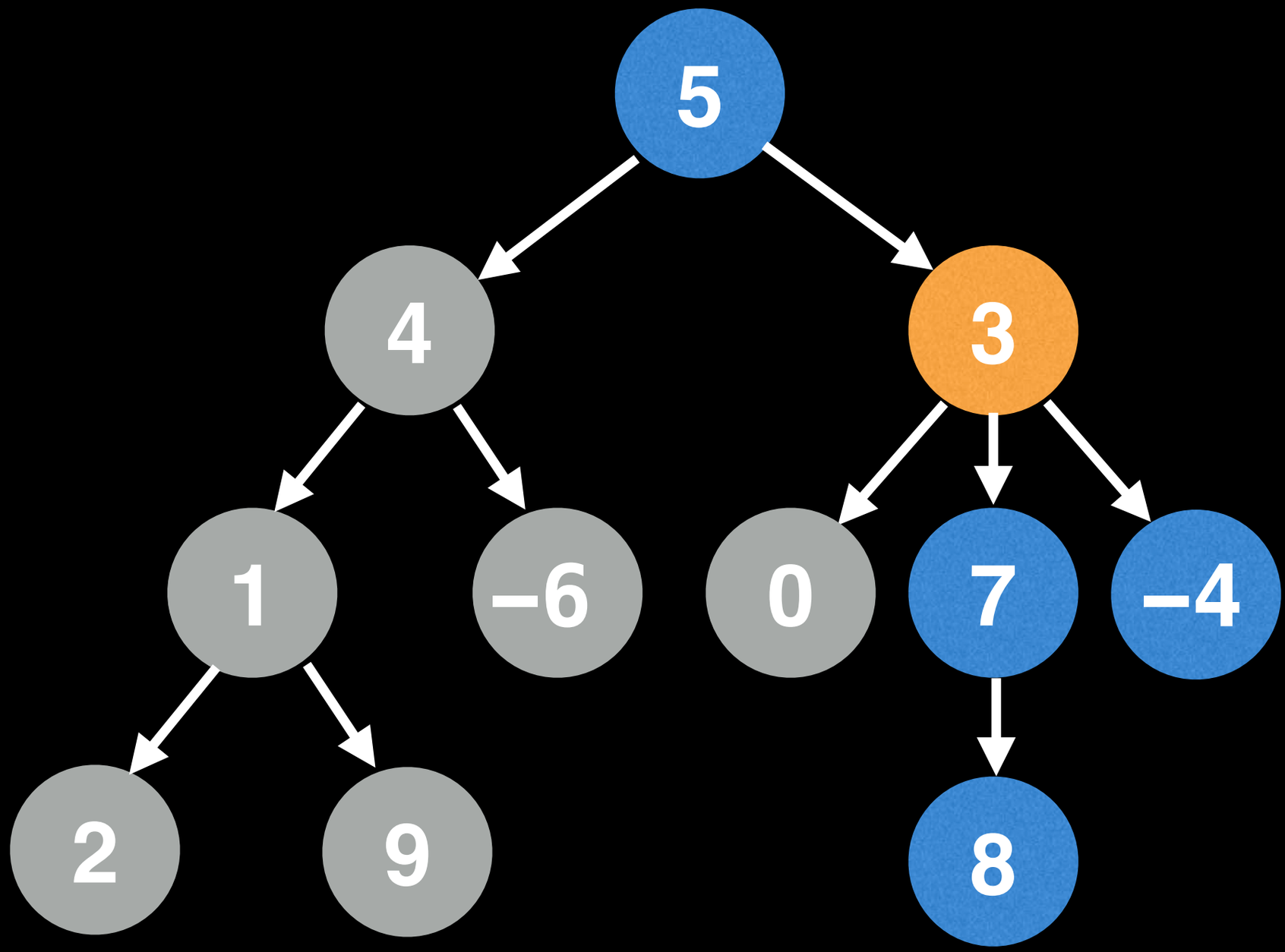
2 + 9 − 6



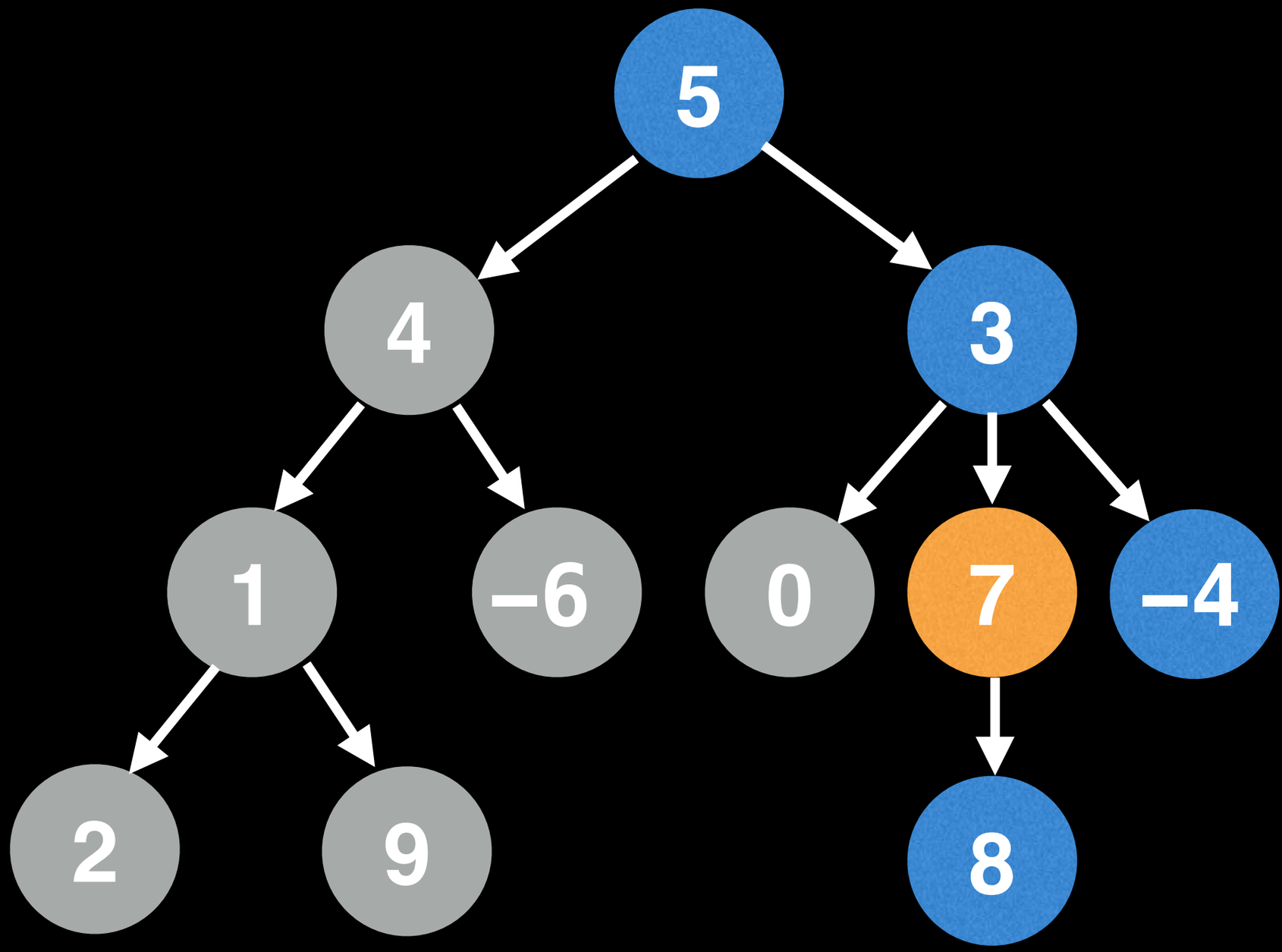
2 + 9 − 6



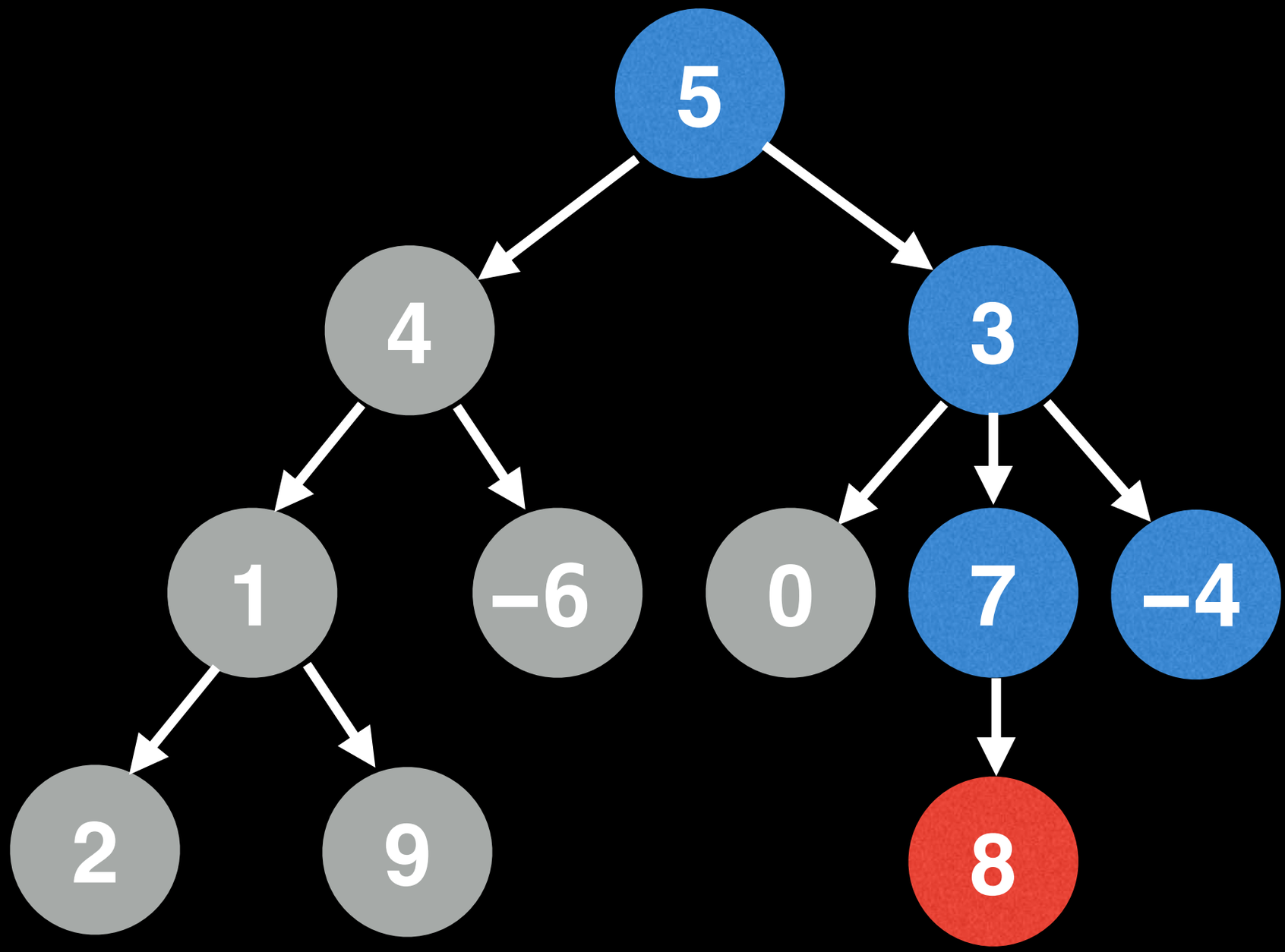
2 + 9 − 6



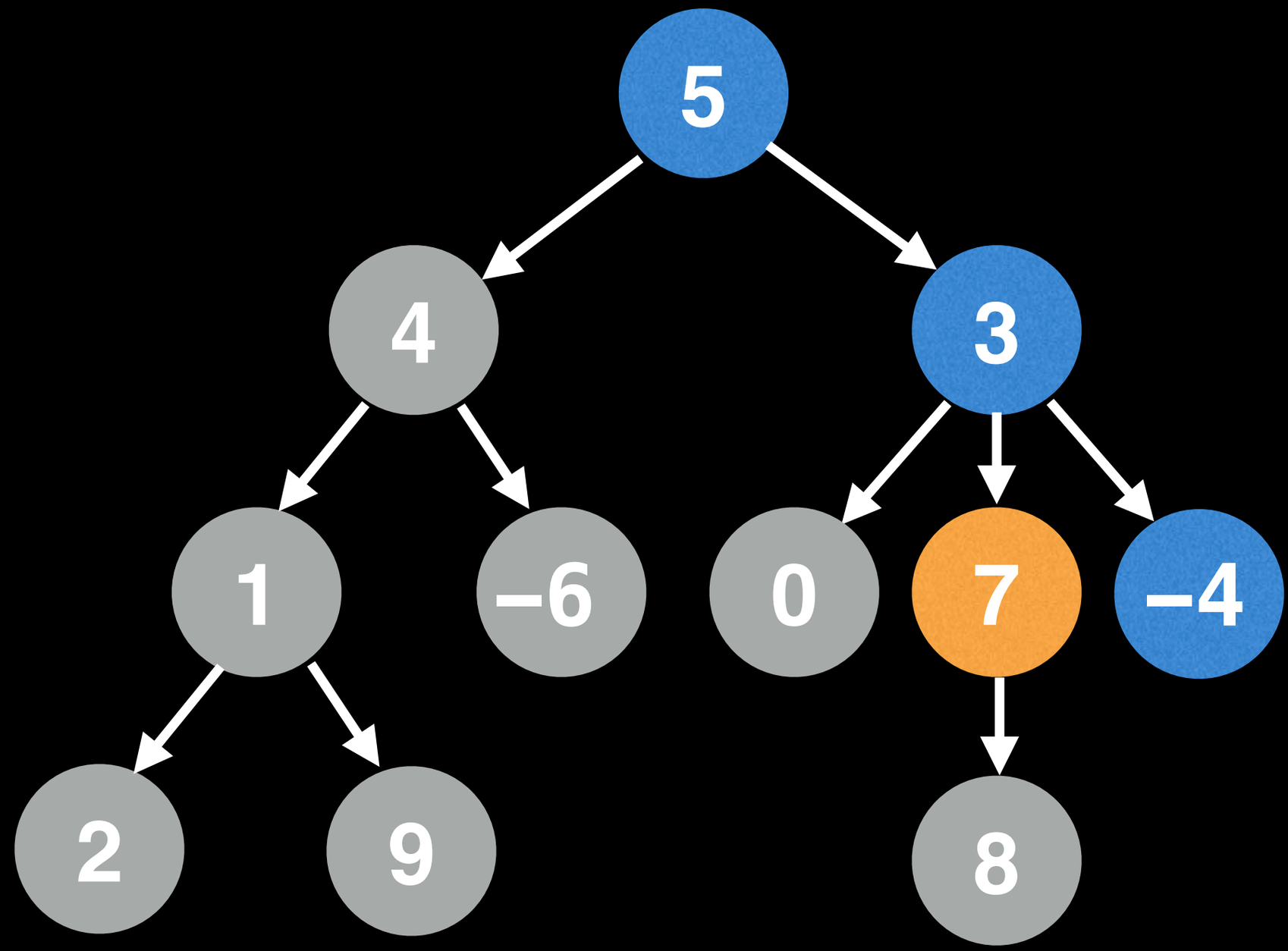
2 + 9 − 6 + 0



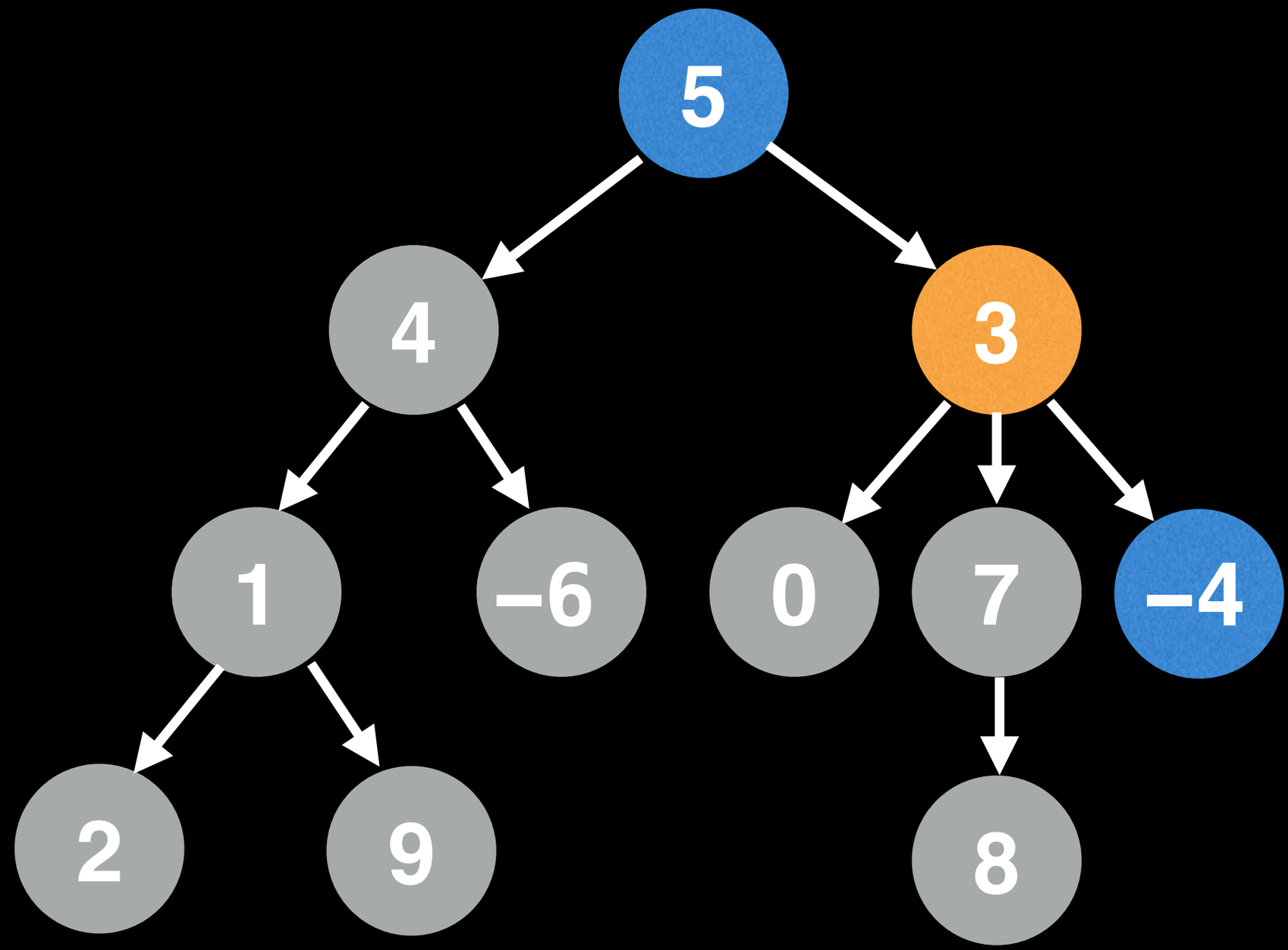
2 + 9 − 6 + 0



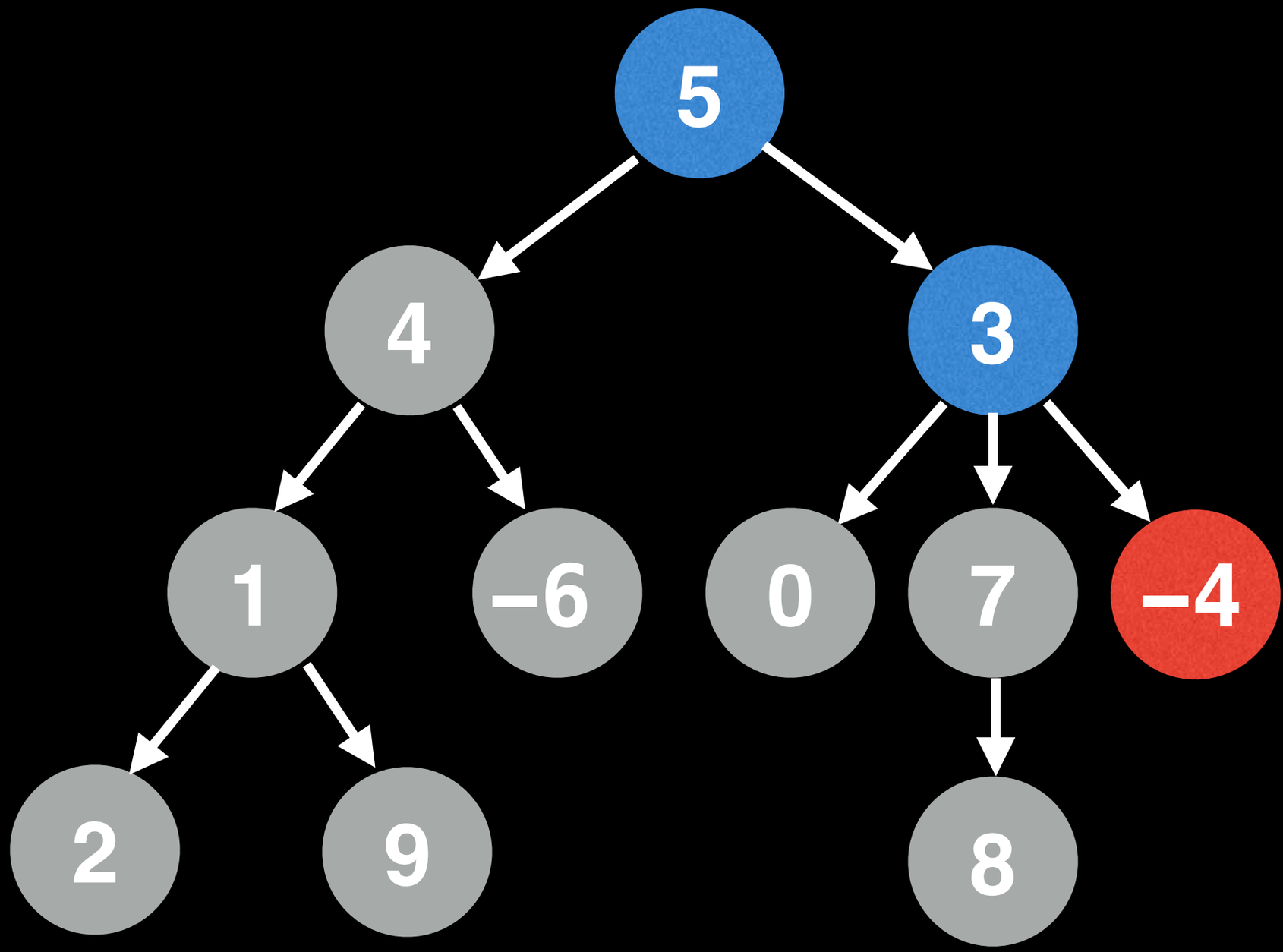
2 + 9 − 6 + 0



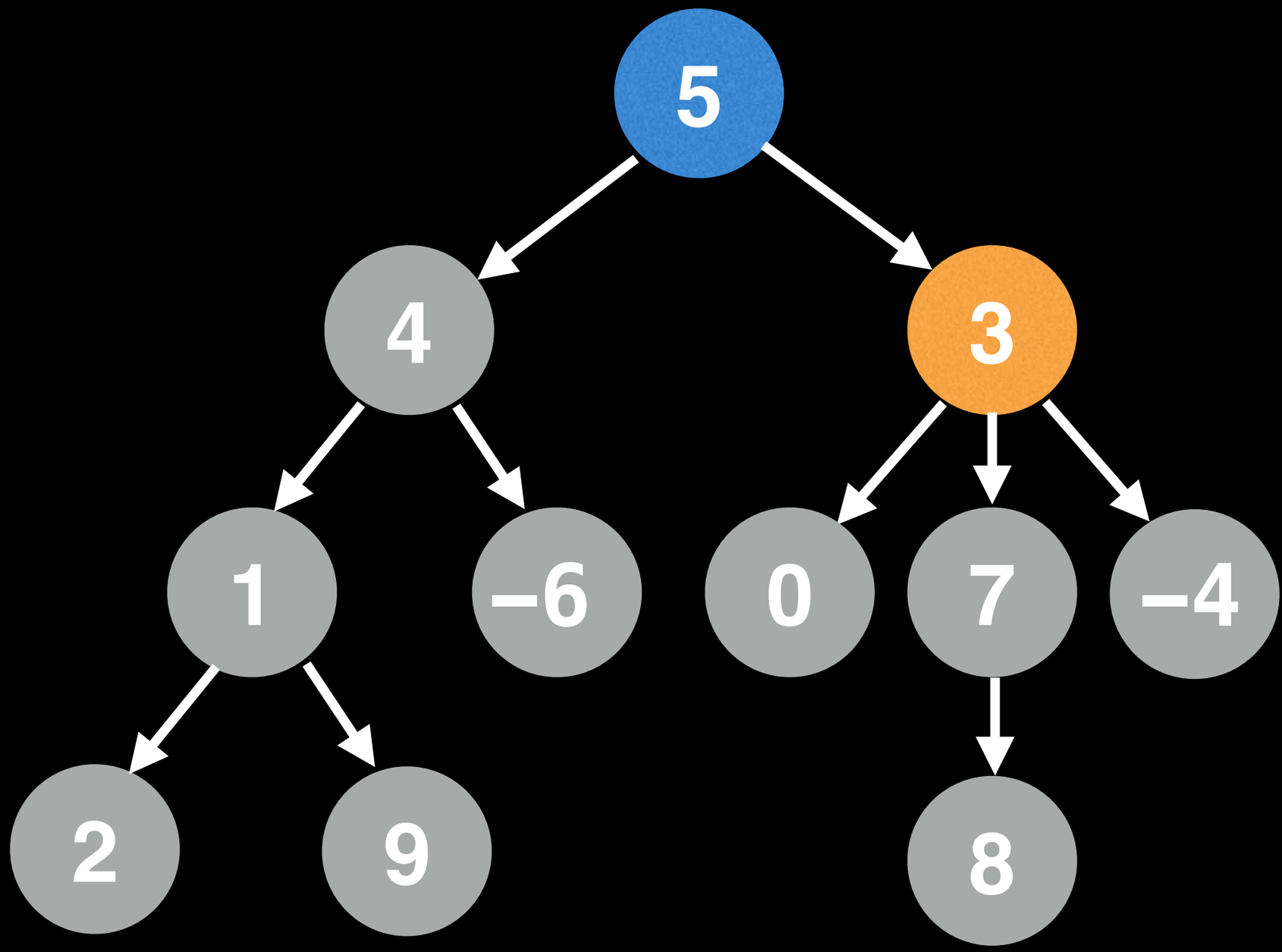
2 + 9 − 6 + 0 + 8



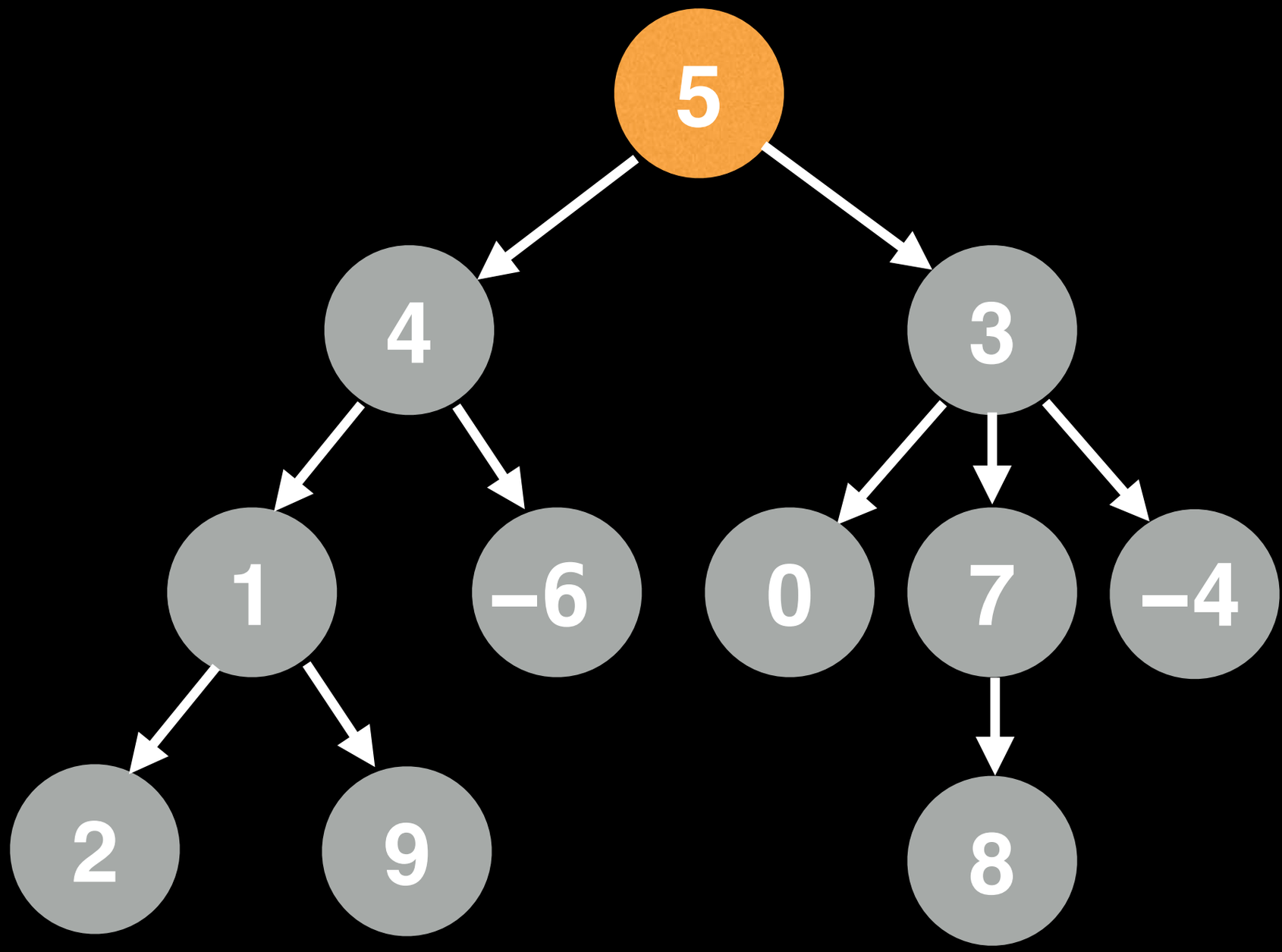
2 + 9 − 6 + 0 + 8



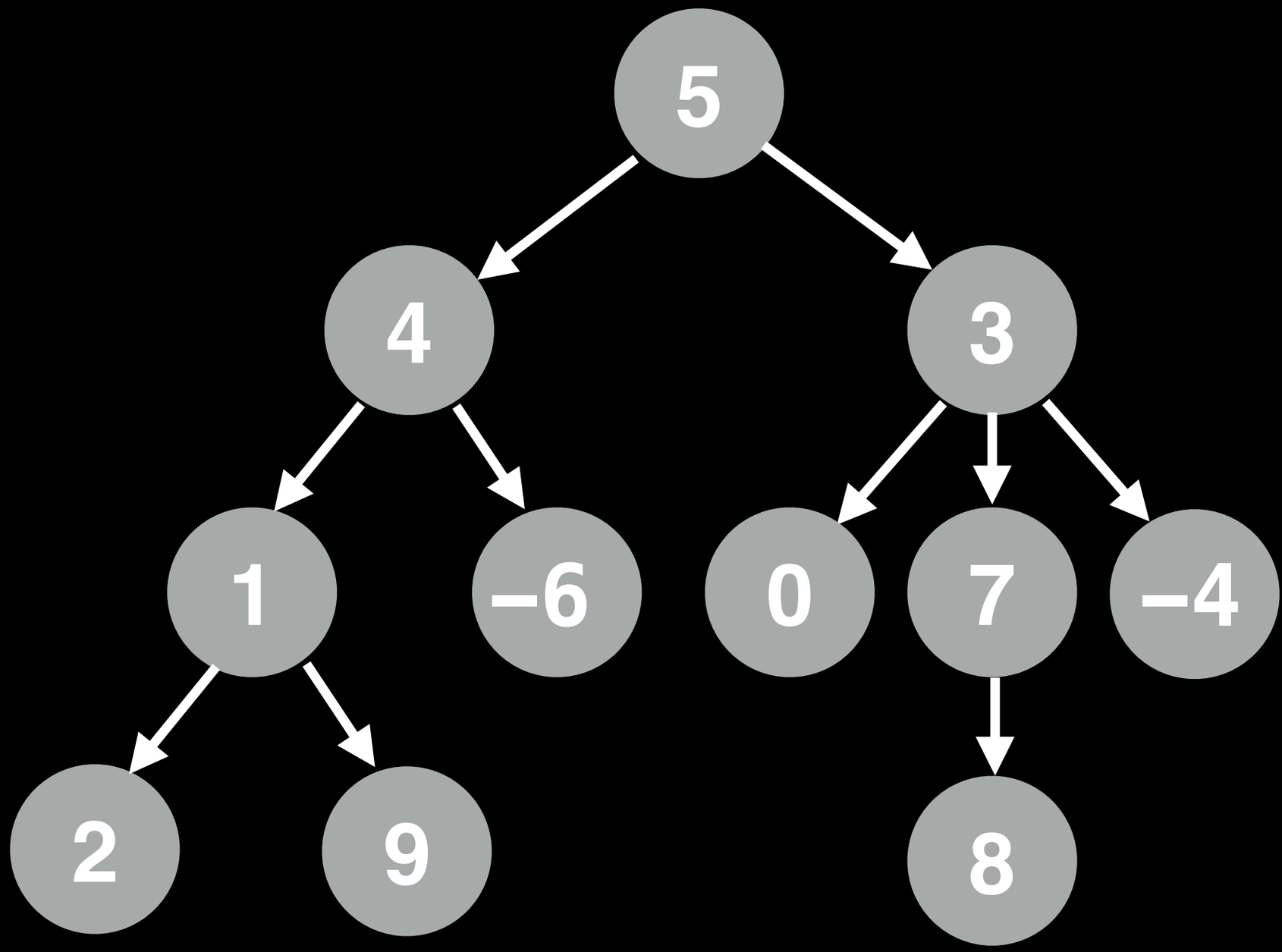

```
2 + 9 − 6 + 0 + 8
```



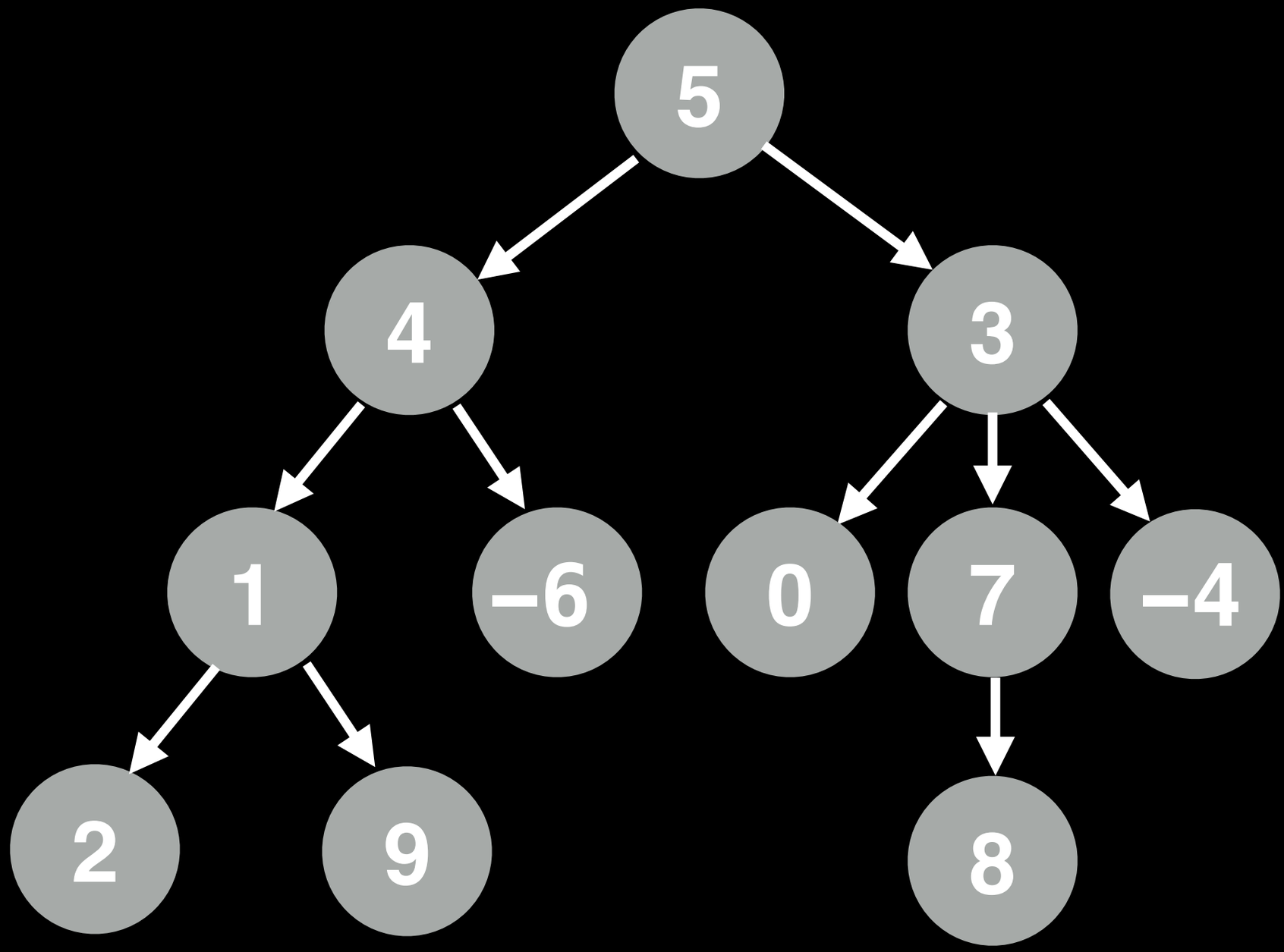
2 + 9 − 6 + 0 + 8 − 4



2 + 9 − 6 + 0 + 8 − 4



2 + 9 − 6 + 0 + 8 − 4



$$2 + 9 - 6 + 0 + 8 - 4 = 9$$

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
      total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```

```
# Sums up leaf node values in a tree.
# Call function like: leafSum(root)
function leafSum(node):
  # Handle empty tree case
  if node == null:
    return 0
  if isLeaf(node):
    return node.getValue()
  total = 0
  for child in node.getChildNodes():
    total += leafSum(child)
  return total

function isLeaf(node):
  return node.getChildNodes().size() == 0
```