

Learning Outcomes

Upon completing this assignment, students should be able to

- Trace Java function calls with *activation records*, including **recursive** calls.

Tracing Code

- Draw the *calling stack* just before the function `second` returns.

```

1 void first(int x) {
2     int a = 2 * x;
3     int b = a - 5;
4     third(a, b);
5 }
6
7 void second(int c) {
8     int d = c * c;
9     System.out.println("d = " + d);
10 }
11
12 void third(int e, int f) {
13     int g = e * f;
14     second(g);
15 }
16
17 public static void main(String [] args) {
18     first(8);
19 }

```

Example:

	main
x 8	first
a 16	
b 11	
ra-main:18	
e 16	third
f 11	
g 176	
ra-first:4	
c 176	second
d 30976	
ra-third:14	

- Draw the calling stack just before `factorial` is called with the value 3.

```

1 int factorial(int n) {
2     if (n > 0)
3         return n * factorial(n-1);
4     return 1;
5 }
6 public static void main(String [] args) {
7     System.out.println(factorial(7));
8 }

```

Solution:

	main
n 7	factorial
ra-main:7	
n 6	factorial
ra-factorial:3	
n 5	factorial
ra-factorial:3	
n 4	factorial
ra-factorial:3	

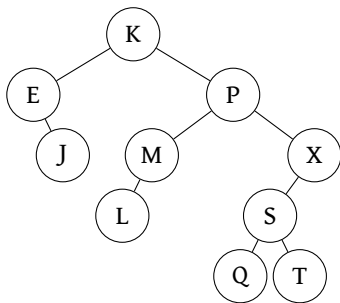
- Draw the calling stack just after the **second** call to `fibonacci(1)` returns. Where in the line (`fibonacci:3`) is it running? What is about to happen and what will the function atop the calling stack return when it is done?

```
1 int fibonacci(int n) {
2   if (n < 2) return n;
3   return fibonacci(n-1) + fibonacci(n-2);
4 }
5 ...
6 fibonacci(3);
7 ...
```

Solution:

	main
n 3	fibonacci
ra-main:6	

4. Draw the calling stack just before the call on the *right* child of L returns from height in the following tree.



```
1 class Tree {
2   private TreeNode root;
3
4   public int height() { return height(root); }
5   private int height(TreeNode curr) {
6     int h = 0;
7     if (curr != null) {
8       h = 1 +
9         Math.max(height(curr.left),
10                  height(curr.right));
11     }
12     return h;
13   }
14 }
```

Solution:

	main
ra-main:...	height()
curr K	height(TreeNode)
ra-height():4	
curr P	height(TreeNode)
ra-height(TreeNode):10	
curr M	height(TreeNode)
ra-height(TreeNode):9	
curr L	height(TreeNode)
ra-height(TreeNode):9	
curr null	height(TreeNode)
ra-height(TreeNode):10	