

Dasar-Dasar Pemrograman 2: Recursion

Tim DDP 2 Fasilkom UI
Correspondence: Fariz Darari (fariz@cs.ui.ac.id)

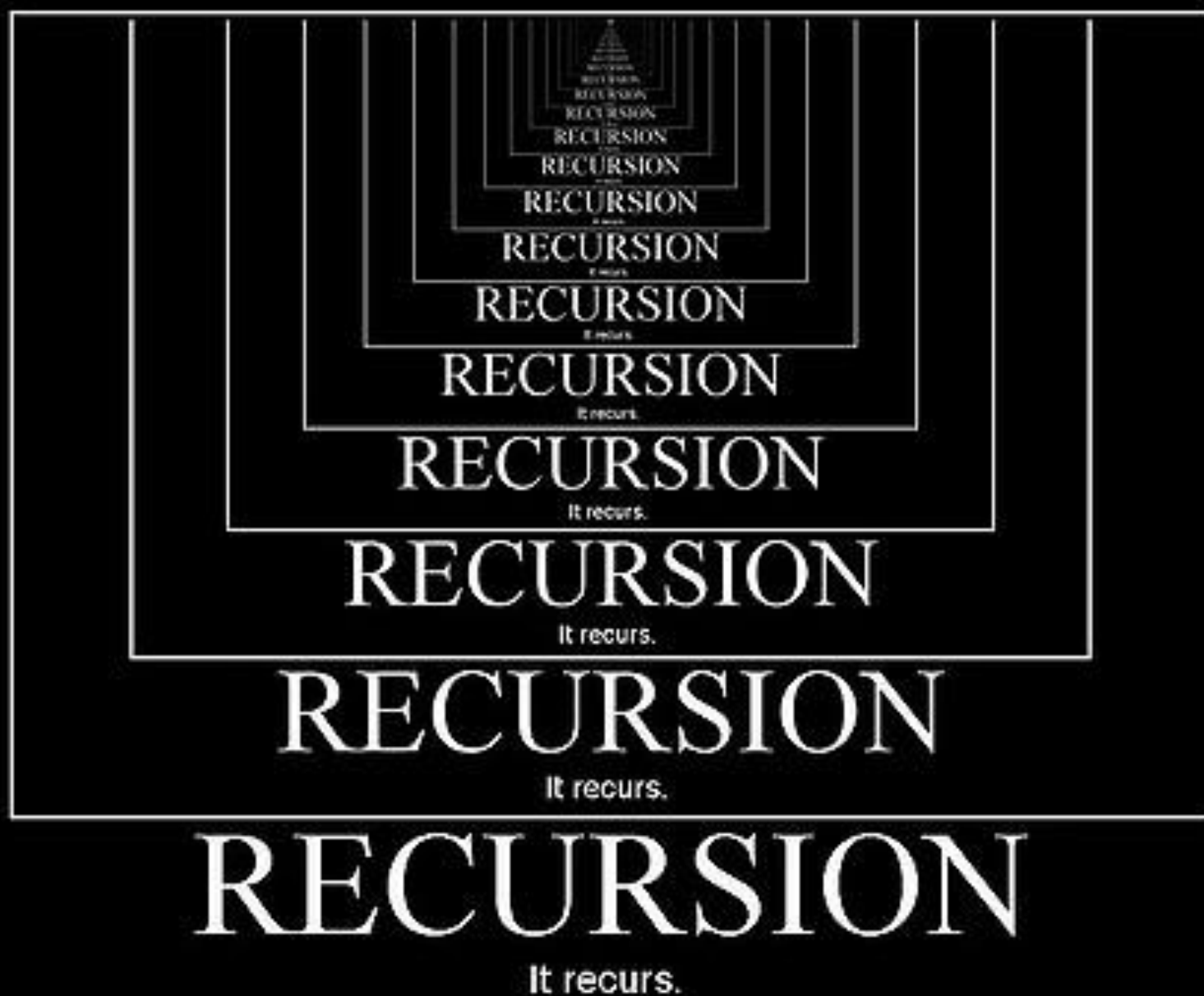
Why?

$$0! = 1$$

$$n! = n \cdot (n - 1)!$$

Try this one:

Create a method to compute the factorial of a non-negative integer n , with **no loops**.



$$0! = 1$$

$$n! = n \cdot (n - 1)!$$

Solution = Recursion

Create a method to compute the factorial of a non-negative integer n , with **no loops**.

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```

$$0! = 1$$

$$n! = n \cdot (n - 1)!$$

Solution = Recursion

Create a method to compute the factorial of a non-negative integer n , with **no loops**.

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```

Recursion = loops with no loops!

$$0! = 1$$

$$n! = n \cdot (n - 1)!$$

Bonus: One-liner solution

Create a method to compute the factorial of a non-negative integer n , with **no loops**.


```
public static int factorialRec(int n) {  
    return n == 0 ? 1 : n * factorialRec(n-1);  
}
```

Short if-else: `condition ? trueCase : falseCase`



Recursion: Base case + recursive case

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```



Recursion: Base case + recursive case

```
public static int factorialRec(int n) {  
    if (n == 0)  Base case  
        return 1;  
    else  
        return n * factorialRec(n-1);  
}
```


Recursion: Base case + recursive case

```
public static int factorialRec(int n) {  
    if (n == 0)  Base case  
        return 1;  
    else  
        return n * factorialRec(n-1);   
}
```

Recursion: Base case + recursive case

```
public static int factorialRec(int n) {  
    if (n == 0)  Base case  
        return 1;  
    else  
        return n * factorialRec(n-1);   
}
```

(!) Make sure argument in recursive case gets simpler!

Quiz time: Iterative sum

Create an iterative method to compute the sum of an array of ints.

Quiz time: Iterative sum

Create an iterative method to compute the sum of an array of ints.

```
public static int sumIter(int[] ints) {  
  
    int sum = 0;  
    for(int i:ints)  
        sum += i;  
    return sum;  
  
}
```

Quiz time: Recursive sum

Create a recursive method to compute the sum of an array of ints.

Quiz time: Recursive sum

Create a recursive method to compute the sum of an array of ints.

```
public static int sumRec(int[] ints) {  
  
    if (ints.length == 0)  
        return 0;  
    else  
        return ints[0] +  
            sumRec(Arrays.copyOfRange(ints, 1, ints.length));  
}
```

Bonus: One-liner recursive sum

Create a recursive method to compute the sum of an array of ints.

One-liner version:

```
public static int sumRec(int[] ints) {  
    return ints.length == 0 ? 0 : ints[0] + sumRec(Arrays.copyOfRange(ints, 1, ints.length));  
}
```

More on `Arrays.copyOfRange()`

- It's a method to copy a subarray given a range.
- The method takes three arguments:
 - `int[]` : array to be copied from
 - `from` : the starting index (inclusive)
 - `to` : the finishing index (exclusive) } **range**
- Example, given that `int[] arr = {5, 1, 2}`:
`Arrays.copyOfRange(arr, 1, arr.length) =`

More on `Arrays.copyOfRange()`

- It's a method to copy a subarray given a range.

- The method takes three arguments:

- `int[]` : array to be copied from
 - `from` : the starting index (inclusive)
 - `to` : the finishing index (exclusive)
- } range

- Example, given that `int[] arr = {5, 1, 2}`:

`Arrays.copyOfRange(arr, 1, arr.length) = [1, 2]`

`Arrays.copyOfRange(arr, 2, 2) =`

More on `Arrays.copyOfRange()`

- It's a method to copy a subarray given a range.

- The method takes three arguments:

- `int[]` : array to be copied from
- `from` : the starting index (inclusive)
- `to` : the finishing index (exclusive) } **range**

- **Example, given that `int[] arr = {5, 1, 2}`:**

```
Arrays.copyOfRange(arr, 1, arr.length) = [1, 2]
```

```
Arrays.copyOfRange(arr, 2, 2) = []
```

```
Arrays.copyOfRange(arr, 0, arr.length) =
```

More on `Arrays.copyOfRange()`

- It's a method to copy a subarray given a range.

- The method takes three arguments:

- `int[]` : array to be copied from
- `from` : the starting index (inclusive)
- `to` : the finishing index (exclusive) } **range**

- **Example, given that `int[] arr = {5, 1, 2}`:**

```
Arrays.copyOfRange(arr, 1, arr.length) = [1, 2]
```

```
Arrays.copyOfRange(arr, 2, 2) = []
```

```
Arrays.copyOfRange(arr, 0, arr.length) = [5, 1, 2]
```

```
Arrays.copyOfRange(arr, 2, 5) =
```

More on `Arrays.copyOfRange()`

- It's a method to copy a subarray given a range.

- The method takes three arguments:

- `int[]` : array to be copied from
 - `from` : the starting index (inclusive)
 - `to` : the finishing index (exclusive)
- } range

- **Example, given that `int[] arr = {5, 1, 2}`:**

```
Arrays.copyOfRange(arr, 1, arr.length) = [1, 2]
```

```
Arrays.copyOfRange(arr, 2, 2) = []
```

```
Arrays.copyOfRange(arr, 0, arr.length) = [5, 1, 2]
```

```
Arrays.copyOfRange(arr, 2, 5) = [2, 0, 0]
```

Quiz time: Recursive palindrom checking

Create a recursive method to check if a string is a palindrom.

Example:

```
System.out.println(isPalin("ada ada"));  
System.out.println(isPalin("malam"));  
System.out.println(isPalin("o"));  
System.out.println(isPalin("lalala"));
```

Quiz time: Recursive palindrom checking

Create a recursive method to check if a string is a palindrom.

Example:

```
System.out.println(isPalin("ada ada"));  
System.out.println(isPalin("malam"));  
System.out.println(isPalin("o"));  
System.out.println(isPalin("lalala"));
```

Output:

```
true  
true  
true  
false
```

Quiz time: Recursive palindrom checking

```
public static boolean isPalin(String s) {  
    if(s.length() == 0 || s.length() == 1)  
        return true;  
    else  
        return s.charAt(0) == s.charAt(s.length()-1) &&  
            isPalin(s.substring(1, s.length()-1));  
}
```

Don't do this anywhere!

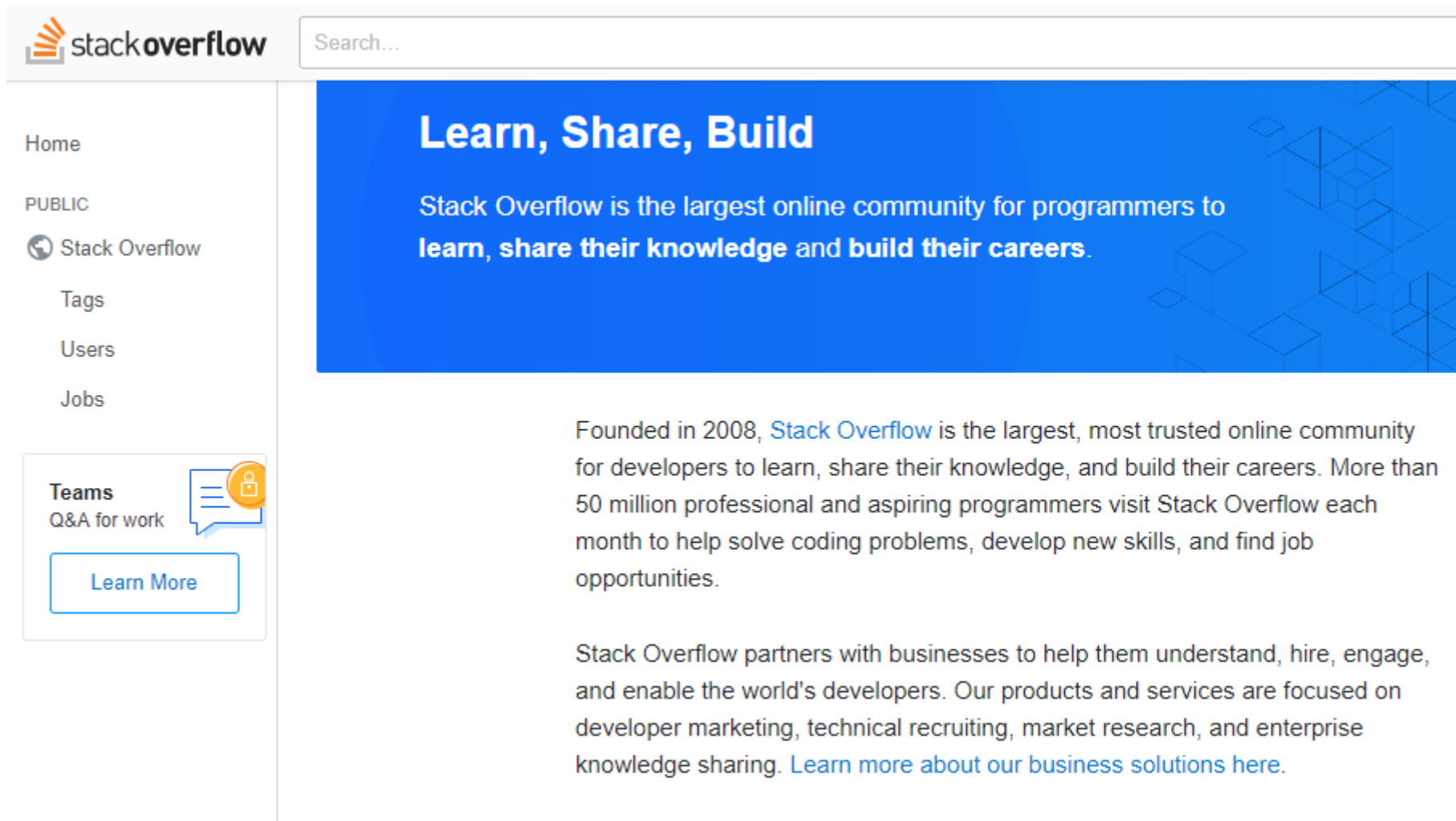
```
public static void forever() {  
    forever();  
}
```


Don't do this anywhere!

```
public static void forever() {  
    forever();  
}
```

(X) When run, it throws `java.lang.StackOverflowError`!

Talking about stackoverflow..



The image shows a screenshot of the Stack Overflow website. At the top left is the Stack Overflow logo. To its right is a search bar with the text "Search...". Below the logo is a vertical navigation menu with the following items: "Home", "PUBLIC", "Stack Overflow" (with a lock icon), "Tags", "Users", and "Jobs". Below the menu is a "Teams" section with the text "Q&A for work" and a "Learn More" button. The main content area features a large blue banner with the text "Learn, Share, Build" and "Stack Overflow is the largest online community for programmers to learn, share their knowledge and build their careers." Below the banner is a paragraph of text: "Founded in 2008, Stack Overflow is the largest, most trusted online community for developers to learn, share their knowledge, and build their careers. More than 50 million professional and aspiring programmers visit Stack Overflow each month to help solve coding problems, develop new skills, and find job opportunities." Below this paragraph is another paragraph: "Stack Overflow partners with businesses to help them understand, hire, engage, and enable the world's developers. Our products and services are focused on developer marketing, technical recruiting, market research, and enterprise knowledge sharing. Learn more about our business solutions here."

stackoverflow

Search...

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work

Learn More

Learn, Share, Build

Stack Overflow is the largest online community for programmers to **learn, share their knowledge** and **build their careers**.

Founded in 2008, [Stack Overflow](#) is the largest, most trusted online community for developers to learn, share their knowledge, and build their careers. More than 50 million professional and aspiring programmers visit Stack Overflow each month to help solve coding problems, develop new skills, and find job opportunities.

Stack Overflow partners with businesses to help them understand, hire, engage, and enable the world's developers. Our products and services are focused on developer marketing, technical recruiting, market research, and enterprise knowledge sharing. [Learn more about our business solutions here.](#)

Talking about stackoverflow..

stackoverflow [java]

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work
Learn More

Questions tagged [java] [Ask Question](#)

Java (not to be confused with JavaScript or JScript or JS) is a general-purpose, statically typed, object-oriented programming language designed to be used in conjunction with the Java Virtual Machine (JVM). "Java platform" is the name for a computing system that has installed tools for developing and running Java programs. Use this tag for questions referring to the Java programming language or Java platform tools.

[Learn more...](#) [Top users](#) [Synonyms \(10\)](#) [java jobs](#)

1,517,718 questions

info Newest **1B** Featured Frequent Votes Active unanswered

22622 votes [Why is it faster to process a sorted array than an unsorted array?](#)

23 answers

java c++ performance optimization branch-prediction

asked Jun 27 '12 at 13:51

GManNickG
301k ● 40 ● 417 ● 511

6304 votes [Why is subtracting these two times \(in 1927\) giving a strange result?](#)

9 answers

java date timezone

asked Jul 27 '11 at 8:15

Freewind
77.5k ● 115 ● 313 ● 563

5751 votes [Is Java "pass-by-reference" or "pass-by-value"?](#)

Talking about stackoverflow..

stack overflow Search...

```
36 if (dev.isBored() || job.sucks()) {
37     searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 // A career site that's by developers, for developers.
```

stack overflow JOBS Get started

How do you find the sum of all the numbers in an array in Java?

105

java arrays sum

share improve this question

edited Feb 24 '16 at 19:13 james.garriss 8,587 5 66 83

asked Dec 29 '10 at 0:35 TomLisankie 1,321 5 20 26

13 [Write your own, the code to do it is 2-3 lines long.](#) – wkl Dec 29 '10 at 0:39

1 Unfortunately the above (and following) "answers" are "The Java Way" :-/ You could use the [Functional Java library](#), but it is so cumbersome to deal with the Java syntax. – user166390 Dec 29 '10 at 2:01

1 I know this question is extremely old, but the [answer by msayag](#) below seems like it should be marked as the accepted answer. – Matsu Q. Sep 16 '16 at 19:41

add a comment

23 Answers active oldest votes

In [java-8](#) you can use streams:

Quiz time: Guess what the code does!

```
public static void cd(int n, String m) {  
  
    if (n == 0) {  
        System.out.println(m);  
    } else {  
        System.out.println(n);  
        cd(n-1, m);  
    }  
  
}
```

Quiz time: Guess what the code does!

```
public static void cd(int n, String m) {  
  
    if (n == 0) {  
        System.out.println(m);  
    } else {  
        System.out.println(n);  
        cd(n-1, m);  
    }  
  
}
```

Answer:

Count down from n to 1, then print message m

Bonus: Countdown with timer!

```
public static void cd(int n, String m) {  
  
    if (n == 0) {  
        System.out.println(m);  
    } else {  
        System.out.println(n);  
        try {  
            Thread.sleep(1000); // sleep for 1000 ms  
        } catch (InterruptedException e) {}  
        cd(n-1, m);  
    }  
  
}
```

Quiz time: What's the output of `cd2(5, "HBD!")` !

```
public static void cd2(int n, String m) {  
  
    if (n == 0) {  
        System.out.println(m);  
    } else {  
        cd2(n-1, m);  
        System.out.println(n);  
    }  
  
}
```


Quiz time: What's the output of `cd2(5, "HBD!")` !

```
public static void cd2(int n, String m) {  
  
    if (n == 0) {  
        System.out.println(m);  
    } else {  
        cd2(n-1, m);  
        System.out.println(n);  
    }  
  
}
```

Output

HBD!

1

2

3

4

5

Decimal-to-binary conversion

For example, to convert 23 to its binary representation, you first repeatedly divide 23 by 2:

23 / 2	is 11	remainder 1
11 / 2	is 5	remainder 1
5 / 2	is 2	remainder 1
2 / 2	is 1	remainder 0
1 / 2	is 0	remainder 1

**Then, you read these remainders from bottom to top.
So, 23 in binary is 10111.**

Quiz time: Recursive decimal-to-binary conversion

Given a positive, base-10 integer, print its binary form.

PS: Base case is when the integer is 0, print nothing.

```
public static void printDecToBin(int dec) {  
    // complete the code..  
  
}
```

Quiz time: Recursive decimal-to-binary conversion

Given a positive, base-10 integer, print its binary form.

PS: Base case is when the integer is 0, print nothing.

```
public static void printDecToBin(int dec) {  
  
    if (dec > 0) {  
        printDecToBin(dec / 2);  
        System.out.print(dec % 2);  
    }  
  
}
```

Quiz time: Fibonacci sequence

$$\text{fib}(1) = 1$$

$$\text{fib}(2) = 1$$

$$\text{fib}(n) = \text{fib}(n - 1) + \text{fib}(n - 2)$$

Quiz time: Fibonacci sequence

$$\text{fib}(1) = 1$$

$$\text{fib}(2) = 1$$

$$\text{fib}(n) = \text{fib}(n - 1) + \text{fib}(n - 2)$$

```
public static int fib(int n) {  
  
    if (n == 1 || n == 2)  
        return 1;  
    else  
        return fib(n-1) + fib(n-2);  
  
}
```

Revisiting this factorial recursion

Try it out with $n = 13$, what happens?

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```

Revisiting this factorial recursion

Try it out with $n = 13$, what happens?

It returns 1,932,053,504

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```


Revisiting this factorial recursion

Try it out with $n = 13$, what happens?

It returns 1,932,053,504 (should've been 6,227,020,800). **But why?**

```
public static int factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```

Revisiting this factorial recursion

Try it out with $n = 13$, what happens?

It returns 1,932,053,504 (should've been 6,227,020,800). **But why?**

```
public static int factorialRec(int n) {  
    // Max value of int is 2,147,483,647
```

```
    if (n == 0)
```

```
        return 1;
```

```
    else
```

```
        return n * factorialRec(n-1);
```

```
}
```

Revisiting this factorial recursion

Solution:

```
public static long factorialRec(int n) {  
  
    if (n == 0)  
        return 1;  
    else  
        return n * factorialRec(n-1);  
  
}
```

Tips: Avoiding infinite recursion

Most of the time, an infinite recursion will cause the program to throw a `StackOverflowError`. But if the program is slow, it may take a long time to fill the stack.

If you know which method is causing an infinite recursion, check that there is a base case. There should be some condition that makes the method return without making a recursive invocation. If not, you need to rethink the algorithm and identify a base case.

If there is a base case, but the program doesn't seem to be reaching it, add a print statement at the beginning of the method that displays the parameters. Now when you run the program you see a few lines of output every time the method is invoked, and you can see the values of the parameters. If the parameters are not moving toward the base case, you might see why not.



**THANK
YOU**

Credits: Chapter 5 & 6 of Think Java book by Allen Downey and Chris Mayfield