I have an array where every number in the range $1...n$ appears once except for one number which appears twice.

Write a function for finding the number that appears twice.

Gotchas

We can do this with $O(1)$ additional memory.

Breakdown

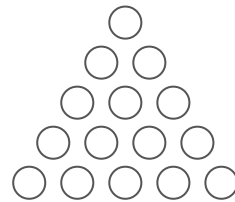To avoid using up extra memory space, lets use some math!

Solution

First, we sum all numbers $1...n$. We can do this using the equation:

$$\frac{n^2+n}{2}$$

because the numbers in $1...n$ are a triangular series

A **triangular series** is a series of numbers where each number could be the row of an equilateral triangle.

So 1, 2, 3, 4, 5 is a triangular series, because you could stack the numbers like this:

Their sum is 15, which makes 15 a **triangular number**.

A triangular series *always* starts with 1 and increases by 1 with each number.

Since the only thing that changes in triangular series is the value of the highest number, it's helpful to give that a name. Let's call it **n**.

```Java
// n is 8
1, 2, 3, 4, 5, 6, 7, 8
```

Triangular series are nice because no matter how large **n** is, it's always easy to find the total sum of all the numbers.

Take the example above. Notice that if we add the first and last numbers together, and then add the second and second-to-last numbers together, they have the same sum! This happens with *every* pair of numbers until we reach the middle. If we add up all the pairs of numbers, we get:

```Java
1 + 8 = 9
2 + 7 = 9
3 + 6 = 9
4 + 5 = 9
```

This is true for *every triangular series*:

1. **Pairs of numbers** on each side will always **add up to the same value**
2. That value will always be **1 more than the series'** n

This gives us a pattern. Each pair's sum is $n + 1$, and there are $\frac{n}{2}$ pairs. So our total sum is:

$$(n + 1) * \tfrac{n}{2}$$

Or:

$$\frac{n^2 + n}{2}$$

Ok, but does this work with triangular series with an *odd* number of elements? Yes. Let's say n is 5. So if we calculate the sum by hand:

$$1 + 2 + 3 + 4 + 5 = 15$$

And if we use the formula, we get the same answer:

$$\tfrac{5^2 + 5}{2} = 15$$

One more thing:

What if we know *the total sum*, but we *don't* know the value of n?

Let's say we have:

$$1 + 2 + 3 + \ldots + (n-2) + (n-1) + n = 78$$

No problem. We just use our formula and set it equal to the sum!

$$\frac{n^2+n}{2} = 78$$

Now, we can rearrange our equation to get a *quadratic equation* (remember those?)

$$n^2 + n = 156$$

$$n^2 + n - 156 = 0$$

Here's the quadatic formula:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If you don't really remember how to use it, that's cool. You can just use an online calculator. We don't judge.

Taking the positive solution, we get $n = 12$.

So for a triangular series, remember—the total sum is:

$$\frac{n^2+n}{2}$$

.

Second, we sum all numbers in our input array, which should be the same as our other sum but with our repeat number added in twice. So the difference between these two sums is the repeated number!

Complexity

$O(n)$ time. We can sum all the numbers $1...n$ in $O(1)$ time using the fancy formula, but it still takes $O(n)$ time to sum all the numbers in our input array.

$O(1)$ additional space. The only additional vars we need to allocate are numbers.

Bonus

If our array contains huge numbers or is really long, our sum might be so big it causes an integer overflow . What are some ways to protect against this?