

String Rotation:

Assume you have method `isSubstring`, which checks if one word is a substring of another. Given strings `s1` and `s2`, write code to check if `s2` is a rotation of `s1`, using only one call to `isSubstring`.

Example:

"waterbottle" → rotation → "erbottlewat"

- A rotation seems to be defined as letters of the word in the same order, just shifted to different positions, while maintaining that order

First thoughts:

in the example, "waterbottle" is broken down such that:

waterbottle → erbottlewat

- is "wat" the substring we should call `isSubstring` on?
- order matters, so we can't sort the strings
- Concatenating `s1` & `s2`:

erbottlewaterbottlewat
 $s1 \times 2$

bottlewaterbottlewater
 $s2 \times 2$

- by concatenating either string to itself, it will contain the other string within itself, which we can then call `isSubstring` for.

Implementation

```
function isRotation(s1, s2) {
```

```
  if (s1.length === s2.length && s1.length !== 0) {
```

```
    s1 = s1 + s1;
```

```
    return isSubstring(s2, s1);
```

```
  }
```

```
return false;
```

```
}
```

Complexity

Time: From the use of `substring` - $O(A+B)$ where A & B are the lengths of the strings being compared

space: $O(m)$ where m is $\frac{1}{2}$ the length of the concatenated string, we need more space due to the concatenation