CTCI 1.9: String Rotation

October 26, 2017

Question

Assume you have a method isSubstring which checks if one word is a substring of another. Given two strings, s1 and s2, write code to check if s2 is a rotation of s1 using only one call to isSubstring (e.g., ”waterbottle” is a rotation of ”erbottlewat”).

For example:

Explanation and Algorithm

The first step would be to figure out the rotation point (aka. at which point can you take the s1, split it into two parts (call them x and y, and rearrange those points to firm the s2). The idea would be that the strings are rotations of each other if s1=xy, s2=yx. Regardless of where this rotation point is, we can see that yx is a substring of xyxy. Essentially, this means that s1 is a substring of s2.

Hints

1. What are some conditions that no matter what, the strings can't be rotations of each other? Think of length.
2. Recall in the explanation and algorithm, s2 must be a substring of s1+s1. In order for the strings to be rotations of each other, they must fulfill these conditions.

Code

```
/*Answer 1*/

boolean isRotation(String s1, String s2){
  int length = s1.length();
```

```
   if(length == s2.length() && length > 0){
      String s1s1 = s1 + s1;
        return isSubstring(s1s1, s2);
    }
   return false;
}
```

Run time analysis

Assuming that isSubstring runs of $O(a + b)$ time where a = length of s1 and b = length of s2, runtime of isRotation is $O(n)$.