**Cyclic rotation of a string.** A string s*s* is a cyclic rotation of a string *t* if *s* and *t* have the same length and *s* consists of a suffix of *t* followed by a prefix of *t*. For example, "winterbreak" is a cyclic rotation of "breakwinter" (and vice versa). Design a linear-time algorithm to determine whether one string is a cyclic rotation of another.

**Tandem repeat.** A tandem repeat of a base string *b* within a string s*s* is a substring of *s* consisting of at least one consecutive copy of the base string *b*. Given *b* and *s*, design an algorithm to find a tandem repeat of *b* within *s* of maximum length. Your algorithm should run in time proportional to *M*+*N*, where *M* is length of *b* and *N* is the length *s*.

For example, if *s* is "abcabcababcaba" and *b* is "abcab", then "abcababcab" is the tandem substring of maximum length (2 copies)

**Longest palindromic substring.** Given a string *s*, find the longest substring that is a palindrome in expected linearithmic time.

Signing bonus: Do it in linear time in the worst case.