CS224 Fall 2011, HW1 Due Thursday Sept. 29 (please put your answers in the homework box for the class in room 2131 Kemper hall).

Also, please type your answers. Latex is best, but if you need to, you can type the text and write mathematics by hand. Hard to read homeworks will not be graded.

Problem 1. For a string A of length n, there are n cyclic rotations of A. For example, if the string is A = abba, the 4 cyclic rotations are: abba, bbaa, baab, aabb. If we lexicographically order those four strings, they are ordered aabb, abba, baab, bbaa. The last character of each string, in the lexicographic order of the strings, concatenated together creates the string A' = baba. In general, given string A, create the string A' of length n by forming the n cyclic rotations of A, sorting them in lexicographic order, extracting and concatenating the last character of each of the strings, in sorted order. Index I points to the position in A' occupied by the last character of the original string. In the example, I = 2. Let (A',I) denote the encoded string and the pointer to the character contributed by the original string. So the encoding of abba is (A',I) = (baba, 2).

Problem 1. Devise a linear time algorithm to reconstruct the original string A from (A',I), and prove that it is correct and that it runs in linear time.

OK - some people may recognize that this transform is called the Burrows-Wheeler transform. You can find the solution to this problem on zillions (literally) of websites. BUT DON'T DO THAT! DON'T EVEN USE THE WEB or other sources TO GET A HINT! Solve the problem YOURSELF!. It is a fun problem and you will enjoy finding the solution. If you just look up the answer, or a hint, you are cheating YOURSELF.

2. Given two strings α and β of the same length n, we want to know if α is a circular shift of β. In other words, are α and β both linearizations of some circular string. For example, $aaba$ is a circular shift of $abaa$. Below is an elegent solution to this problem.

Problems: 2a) Determine if the solution is correct. I'm looking for the big picture here, so try to understand the logic behind the code; if the logic seems to work, but you find a tiny coding bug, fix it before you answer. Don't give an answer like: no, because you have an off-by-one error. Answer no if the basic method really doesn't work.

2b) Prove that your answer in a) is correct.

2c) Analyze the worst-case running time of the method - in this case the measure of running time is the number of character comparisons the algorithm does.

In the pseudo-code below the first entry in the character array is in position one (as sane people should do it), and not in position zero. When comparing characters, the comparison operation $>$ means lexicographically greater.

Determine if α is a circular shift of β.

$x = \alpha\alpha$;
$y = \beta\beta$;
$i = j = 0$;
While $i \leq n$ and $j \leq n$
{
    $k = 1$;
    while $(k \leq n)$ and $(x[i+k] == y[j+k])$ {k++;}
    if $(k > n)$ {print (α is a circular shift of β); break;}
    else if
    $(x[i+k] > y[j+k])$ {i = i+k;}
    else $\{j = j + k;\}$
}
If $(i > n)$ or $(j > n)$ print (α is not a circular shift of β);

For greater clarity, here is Perl code for the method. (You do know Perl, don't you? If not, just pretend you do, and it will all make sense.)

```
#!/pkg/bin/perl -w
# program circ.pl

print "Type in string alpha\n";
chomp( $alpha= <> );
print "Type in string beta\n";
chomp( $beta= <> );
$len1 = length($alpha);
$len2 = length($beta);

if ($len1 != $len2) {
```

```
   print "The two strings have unequal lengths $len1 and $len2\n"; }
else {
  $x = $alpha . $alpha;
  $y = $beta . $beta;
  $i = 0;
  $j = 0;
  while (($i <= $len1) && ($j <= $len2))
  {
    $k = 1;
    while (($k <= $len1) &&
       ($c1 = substr($x, $i+$k, 1)) eq ($c2 = substr($y, $j+$k, 1)))
    {$k++}
    if ($k > $len1) {
       print "alpha is a circular shift of beta\n";
       last;
    }
    elsif ($c1 gt $c2) {$i = $i + $k}
     else {$j = $j + $k}
 }

if (($i > $len1) || ($j > $len1)) {
       print "alpha is not a circular shift of beta\n"; }
}
```

OK, Again, this is a problem I have used in prior classes, and you probably can find an answer by searching for it or asking the right student, BUT DON'T! Solve the problem yourself for the benefit and enjoyment of doing it. You can thank me later.