CS 122A fall 2010, HW 7 Due November 24. The extension date on this is Nov. 29. So you have a lot of time for this, but parts of it are hard, so don't wait until the last minute to do it.

1. Use the Z-algorithm to compute the Z values for the string $S = ababbbabababc$. You must state exactly what the Z-algorithm would do for each position i, i.e., which case it is in, and what all the relavent variables are etc.. You may use the version of the Z-algorithm we developed in class, or the version in the posted notes, but you need to state which version you are using.

2. I conjecture that for any i where $Z_i(S) > 0$, $Z_{i+1}(S) = \min[Z_2(S), Z_i(S) - 1]$. If you agree, give an explanation, and if you disagree, give a counter-example. If you think it is not true but it can be modified simply so that it is true, explain that also. The main point of the conjecture is that for any position i where $Z_i(S) > 0$, $Z_{i+1}(S)$ can be computed in constant time, without any character comparisons.

To solve the problem, it is useful to recall and to understand that $Z_2(S)$ is equal to the number of consecutive characters at the start of S that equal the first character in S, minus one.

As extra credit, if you think the conjecture is true, can you use it to reduce the upper bound on the number of character comparisons made by the Z-algorithm to something below $2|S|$?

3. Define $Q_i(S)$ as the length of the longest substring that starts at position i and matches a suffix of S starting at some position not equal to i. For example, in the string $S = rzbaaaxpdqbaaax$, $Q_3 = 5$.

Show how to compute all the Q values (that is for all the i) in $O(|S|)$ time. This is not as obvious as it looks. It is not true that you just reverse S and compute the Z values. But if you try that idea and see why it fails, you will probably next see the right idea.

4. Let T be a text string of length m and let Π be a *multi-set* of n characters (a multi-set is a set where an element can occur more than once, but the exact number of occurrances is given). The problem is to find, for each position i in T, the length of the longest substring starting at position i, that can be formed by characters in Π. For example, let $\Pi = \{a,a,b,c\}$ and $T = abaahgcabah$. Then the answer for $i = 1$ is 3 because the substring aba can be formed from the characters in Π. Note that the answer for $i = 1$ is not 4, because that has three copies of character a but Π has only two copies

of character a. The answer for $i = 2$ is also 3, and the answer for $i = 7$ is four.

Give an algorithm to solve this problem that runs in $O(m)$ time. Of course, argue that it is correct and that it does actually run in $O(m)$ time. The solution I have in mind does not involve the Z-algorithm. It is more like a greedy algorithm.