1. We call w primitive if it cannot be written as $w = u^k$ for some $k \geq 2$. Show that w is primitive if and only if if it occurs exactly twice in ww (once as a prefix and once as a suffix).

2. Given a word w, we want to compute, for every $i = 1, 2, \ldots, |w|$, the value of $Pref[i] = \max\{j : w[1..j] = w[i..(i+j-1)]\}$. Design an $O(|w|)$ time algorithm for this problem.

3. To use the good suffix heuristic we need to compute, for every $j = 1, 2, \ldots, m$, the smallest $s > 0$ such that, for every $k = j+1, \ldots, m$, we have $s \geq k$ or $p[k-s] = p[k]$. Show how to compute all such values in $O(m)$ time.

4. To use the strong good suffix heuristic we need to compute, for every $j = 1, 2, \ldots, m$, the smallest $s > 0$ such that, for every $k = j+1, \ldots, m$, we have $s \geq k$ or $p[k-s] = p[k]$, and additionally $s \geq j$ or $p[j-s] \neq p[j]$. Show how to compute all such values in $O(m)$ time.