# BCB 567/CprE 548 Bioinformatics I
# Fall 2007
# Homework 4
# Due Tuesday, October 30

1. (5 points) A string $S$ is called semiperiodic with period $\alpha$ if $S$ is a prefix of $\alpha^k$ (the substring obtained by concatenating $\alpha$ with itself $k$ times) for some positive integer $k$. Give an algorithm to find the shortest period of a string $S$ and compute its run time.

2. (5 points) A substring $\alpha$ of a string $S$ is called a minimal unique substring if it satisfies the following properties:

   - $\alpha$ occurs exactly once in $S$ (uniqueness).
   - All proper prefixes of $\alpha$ occur at least twice in $S$ (minimality).
   - The length of $\alpha$ is at least $l$, for some fixed $l$.

   Give an algorithm to enumerate all the minimal unique substrings of a string $S$.

3. (5 points) Given a string $A$, a set of strings $\mathcal{S}$, and an integer $k$, the primer selection problem is the following: For each position $i$ in $A$, find the shortest substring in $A$ of length at least $k$ that starts in position $i$ and does not occur as a substring of any $s \in \mathcal{S}$. Show how to solve this problem in $O\left(|A| + \Sigma_{s \in \mathcal{S}} |s|\right)$ time.

4. (5 points) **Microarray oligo design problem:** Given a set $S = \{s_1, s_2, \ldots, s_m\}$ of $m$ strings of total length $n$, identify $m$ substrings $\alpha_1, \alpha_2, \ldots, \alpha_m$, such that each substring $\alpha_i$ satisfies the following:

   (a) **Uniqueness:** $\alpha_i$ is a substring of $s_i$, and is not a substring of any other string in $S$.

   (b) **Size:** $\alpha_i$ is the shortest substring of $s_i$ that satisfies (a).

5. (10 points) You are given two strings $S_1$ and $S_2$ and a parameter $k$. A $k$-cover of $S_2$ is a sequence of substrings of $S_1$, each of length $\geq k$, which when concatenated together give $S_2$; i.e., $T_1, T_2, T_3, \ldots, T_l$ is a $k$-cover iff $|T_i| \geq k$, $T_i$ is a substring of $S_1$ $(1 \leq i \leq l)$, and $S_2 = T_1 \parallel T_2 \parallel T_3 \parallel \ldots \parallel T_l$, where $||$ denotes the concatenation operation. The substrings may overlap in $S_1$. Give a linear time algorithm to find $k$-cover, or to determine that no such cover exists.