Introduction to Bioinformatics (autumn 2005)

Excercise 6

Group	time	place
Riikka Kaven	Tuesday 22.11 at 12.15–14.00	BK106

1. Construct an *inverted index* (the table at page 312 in J & P or "hash table" in lecture slides) for 3-mers of `ACCCTAGGTACCCAG` (not required to calculate the actual ℓ-mer to integer mapping here). Find maximal repeats by extending the ℓ-mer repeats. Do you find all maximal repeats this way?

2. Consider the mapping

$$integer(s_1 s_2 \cdots s_\ell) \quad = s_1\sigma^{\ell-1} + s_2\sigma^{\ell-2} + \cdots + s_\ell,$$

   that gives the index of ℓ-mer $S = s_1 s_2 \cdots s_\ell$ in the inverted index. Here the alphabet of S is assumed to be $\Sigma = \{0, 1, 2, \ldots, \sigma - 1\}$ (this is not crucial since any alphabet of size σ can be mapped to Σ, e.g. $\{A, C, G, T\} \rightarrow \{0, 1, 2, 3\}$).

   When building an inverted index, we scan a sequence T from left to right using a window of length ℓ. That is, when the window is at position i we need to compute $integer(t_i t_{i+1} \cdots t_{i+\ell-1})$, at position $i+1$ we need to compute $integer(t_{i+1} t_{i+2} \cdots t_{i+\ell})$, and so on.

   One of the most frequently rediscovered algorithms in the literature is the following *incremental algorithm*: compute $integer(t_{i+1} t_{i+2} \cdots t_{i+\ell})$ *in constant time* given $integer(t_i t_{i+1} \cdots t_{i+\ell-1})$. Rediscover this algorithm.

3. Build the keyword tree for the *suffixes* of `CAACGCAAT`. Convert it into the suffix tree. Consider the paths from root to each internal node of the suffix tree. What is the relationship between these paths and repeats in the sequence?

4. (Problem 9.11 in J & P) Design an efficient algorithm for finding the longest substring shared by two given texts T_1 and T_2.

5. (Problem 9.13 in J & P) Design an efficient algorithm that finds the shortest substring of text T_1 that does not appear in text T_2.

Hint for the two last assignments: Exploit suffix tree of $T_1\$T_2$, where \$ is a special symbol not occuring in T_1 nor in T_2.