Reference Compression

Let $R$ and $S$ be strings over an alphabet $\Sigma$ of length $r$ and $n$, respectively. The string $R$ contains at least one copy of each character in $\Sigma$. The reference parsing of $S$ wrt. $R$ parses $S$ into phrases $p_1, \ldots, p_k$ greedily from left-to-right as follows. Suppose that we have parsed the prefix $S[1, \ell - 1]$ into phrases $p_1, \ldots, p_i - 1$. To obtain $p_i$ we find a longest substring of $S$ starting at position $\ell$ that matches a substring of $R$. The reference compression consists of the string $R$ and the sequence of phrases $p_1, \ldots, p_k$, where each phrase is encoded with its start position and end position in $R$. Thus the total size of the compressed data is $O(r + k)$. Solve the following exercises.

1. Let $R = \text{abbac}$ and $S = \text{abcbbabbaac}$. Show the parsing of $S$ using the bar-notation (as in the slides) along with the encoding of each phrase.

2. Give an efficient encoding algorithm for reference compression.

3. Give an $O(r + k)$ space data structure that supports fast random access queries in $S$ (see weekplan for definition of access queries).