1. Draw the suffix tree for the DNA sequence \texttt{CACTACGTACG}. Include the suffix links from internal nodes as used in Ukkonen’s suffix tree construction algorithm.

2. The algorithm described in class for finding all occurrences of a query string \(Q\) in a suffix tree \(T\) takes time \(O(n + k)\), where \(n\) is the length of \(Q\) and \(k\) is the number of occurrences of \(Q\) in the string encoded by \(T\). Write an algorithm for preprocessing the suffix tree \(T\) in \(O(m)\) time such that finding a single occurrence of \(Q\) in \(T\) takes \(O(n)\) time (hint: label each internal node of the tree with a convenient value).

3. Give updated bounds for the false negative (\(\rho_{fn}\)) and false positive (\(\rho_{fp}\)) rates for LSH-ALL-PAIRS if we form hash functions by picking \(k\) of \(d\) positions at random without replacement.

4. Let \(S\) be the set of all bit strings of length \(\ell\). For \(x \in S\), let \(\text{ones}(x) = \{i : x_i = 1\}\), i.e., the set of all positions in \(x\) that are equal to one. We define the set resemblance, \(s(x, y)\), for \(x, y \in S\) as \(s(x, y) = |\text{ones}(x) \cap \text{ones}(y)| / |\text{ones}(x) \cup \text{ones}(y)|\), i.e., the fraction of positions that are equal to one in both strings out of all positions that are equal to one in either of the strings. For the case of \(x = 0^\ell\) (the all-zero string), define \(s(x, x) = 1\). From this we can define a distance function \(d(x, y) = 1 - s(x, y)\). A great locality-sensitive hash function for this distance measure is called \text{minhash}. Given a random permutation, \(\pi\), of positions \(\{1, 2, \ldots, \ell\}\), \(\text{minhash}_\pi(x) = \arg\min_{i \in \text{ones}(x)}(\pi(i))\), i.e., the index of the first non-zero position in \(x\) in the position ordering given by \(\pi\). Show that that \(\mathbb{P}_\pi[\text{minhash}_\pi(x) = \text{minhash}_\pi(y)] = 1 - d(x, y) = s(x, y), \forall x, y \in S\).