BCB 567/CprE 548
Fall 2007
Homework 3
Due Thursday, October 18

1. (5 points) Draw the suffix tree for the string $xyxyyyxx$. Include suffix links. Use full substring labels for edges (as opposed to a pair of integers).

2. (5 points) Consider the set of all distinct substrings of a string S. This set contains all substrings of S, but each repeat is only represented once. For example, for the string $S = xxyxyy$, this set is $\{x, y, xx, xy, yx, yy, xxy, xyx, yxy, xyy, xxyx, xyxy, yxyy, xxyxy, xyxyy, xxyxyy\}$. The size of this set is $O(n^2)$ and the total length in characters is $O(n^3)$. Therefore if we wished to enumerate the elements of this set we would expect to take $O(n^3)$ time.

   Assume that you are given a suffix tree of S. Describe an algorithm that *counts* the number of distinct substrings of S and runs in $O(n)$ time.

3. (5 points) For the following questions, you are not to consider $|\Sigma|$ to be a constant. Consider the following three ways in which to store the pointers to children of a node in a suffix tree.

   (a) In the first representation, the pointers are stored as an array of size $|\Sigma|$. Thus we allocate space for a pointer even if the corresponding child does not exist (the array entry for a child that does not exist is set to *null*). What is the runtime for finding a pattern P in the string using this formulation? What is the space requirement of the suffix tree?

   (b) In the second representation, the pointers to the children of a node are stored in a linked list. There is exactly one node in the linked list for each child of the node in the suffix tree. What is the runtime for finding a pattern P in the string using this formulation? What is the space requirement of the suffix tree?

   (c) In the third representation, the pointers to the children of a node are stored in a balanced tree. There is exactly one node in the balanced tree for each child of the node in the suffix tree. What is the runtime for finding a pattern P in the string using this formulation? What is the space requirement of the suffix tree?

4. (5 points) In class, we motivate the need of a suffix tree through the example of pattern matching. We saw that, given a suffix tree of a string S and a pattern P, we could enumerate the occurrences of P in S in $O(|P| + C)$ time, where C is the number of occurrences of P.

   A *prefix tree* of a string S is the compacted trie of all prefixes of the string. Can pattern matching be done equally efficiently using a prefix tree of S? Justify your answer.

5. (5 points) Prove or disprove the following statement: The suffix tree of a string and the suffix tree of its reverse have the same number of internal nodes. (Note: To show a statement is false, a counterexample is all that is needed.)

6. (5 points) A string β is called a repeated prefix of a string S if $\beta\beta$ is a prefix of S. That is, $S = \beta\beta\alpha$ for some strings β and α.

   (a) Give a linear time algorithm to find the longest repeated prefix of a string S.

   (b) If $S = \alpha\beta\beta$, then β is a repeated suffix of a string. Assume that you have a method for finding the longest repeated prefix of S in linear time, even if you were unable to solve 6(a). How can you use this method to find the longest repeated suffix of S in linear time?