Homework # 5

Handed out: 9/28/06
Due: 10/3/06

NOTE: you only need to do 3 problems out of the following 4. The fourth depends on material we might not get to in today's class. If we do get to it, you can substitute problem 4 for any of the first 3.

1. Problem 2 from Chapter 6 of Gusfield:
   We first introduced Ukkonen's algorithm at a high level and noted that it could be implemented in $O(m^3)$ time. That time was then reduced to $O(m^2)$ with the use of suffix links and the skip/count trick. An alternative way to reduce the $O(m^3)$ time to $O(m^2)$ (without suffix links or skip/count) is to keep a pointer to the end of each suffix of S[1..i]. Then Ukkonen's high-level algorithm could visit all these ends and create T_{i+1} from T_i in O(i) time, so the entire algorithm would run in $O(m^2)$ time. Explain this in detail.
2. Problem 3 from Chapter 6 of Gusfield:
   The relationship between the suffix tree for a string S and the reverse string S^r is not obvious. However, there's a significant relationship betwen the two trees. Find it, state it, and prove it (by prove I mean: convince me, no need for an extremely formal proof).

   Hint: Suffix links help
3. Problem 3 from Chapter 7 of Gusfield:
   We can define the suffix tree in terms of the keyword tree used in the Aho-Corasick (AC) algorithm. The input to the AC algorithm is a set of patterns P, and the AC tree is a compact representation of those patterns. For a single string S we can think of the n suffixes of S as a set of patterns. Then one can build a suffix tree for S by first constructing the AC tree for those n patterns, and then compressing, in a single edge, any maximal path through nodes with only a single child. If we take this approach, what is the relationship between the failure links used in the keyword tree and the suffix links used in Ukkonen's algorithm? Why aren't suffix trees build this way?
4. Problem 5 from Chapter 6 of Gusfield:
   In trick 3 of Ukkonen's algorithm, the symbol "e" is used as the second index on the label of every leaf edge, and in phase i + 1, the global variable e is set to i + 1. An alternative to using "e" is to set the second index on any leaf edge to m (the total length of S) at the point that the leaf edge is created. In that way, no work is required to update that second index. Explain in detail why this is correct, and discuss any disadvantages there may be in this approach, compared to using the symbol "e".