Algorithms in the Real World (15-853), Fall 07
Assignment #2

Complete all problems. Due October 18.
You are not permitted to look at solutions of previous year assignments. Unfortunately even if I remove them from the web page, Google might have them cached. You can work in groups but must write up the solutions on your own and fully understand them. You can do research on the web or elsewhere but must cite any solutions/partial solutions you find. Also if you use a solution from the web directly, we might deduct some points.

Problem 1: Linear Transform Encoding (10pt)

Consider the following functions defined on $x \in [1, n]$. State which transform (Cosine or Wavelets) gives a better compression in each case and justify in a sentence or two. You can say that they give about the same compression.

(A) $f_1(x) = \begin{cases} 1 & x \geq \frac{n}{4} \\ 0 & \text{otherwise} \end{cases}$

(B) $f_2(x) = x$

(C) $f_3(x) = \begin{cases} 1 & 2i - 1 \leq x \leq 2i \text{ where } i \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}$

(D) $f_4(x) = \begin{cases} f_3(x) & x \leq \frac{n}{2} \\ 0 & \text{otherwise} \end{cases}$

Problem 2: Wavelets (10pt)

Describe how to invert the 5-tap/3-tap wavelet transform used by the lossless version of JPEG2000. In particular describe how given the outputs of the "high pass" filter and the "low pass" filter you can reconstruct the original sequence. You can ignore the boundary conditions.

Problem 3: Suffix Trees (20pt)
Show the suffix tree for the string `bcabccabc$` including all suffix links. Now give the sequence of i and j values for the suffix tree algorithm described in class (also in the chapter of Gusfield's book available on line) for this string. For the example in the slides (`xabxac`) the answer would be.

i	j	rule
1	1	2
2	2	2
3	3	2
4	4	3
5	4	3
6	4	2
6	5	2
6	6	2

Note that there is an implicit application of rule 3 when rule 2 is applied and $i = j$. Including these implicit applications there should be exaclty n applications of each of rule 2 and rule 3 for a string of length n.

Problem 4: Minimum Cover (10pt)

Given strings A and B, a minimum cover of A by B is a decomposition $A = w_1 w_2 \cdots w_k$ where each w_i is a substring of B and k is minimum. Show how to compute a minimum cover (given that one exists) in $O(|A| + |B|)$ time.

Problem 5: Sliding Window Lempell Ziv (20pt)

Please describe how to use suffix trees to implement LZ77 (sliding windows Lempell Ziv) on a string S of length n, with dictionary size d and buffer size b. You can use $O(n)$ time and space. Note that LZ77 can only report a match within the window.