COS 423 Problem Set 1 (six problems, including one extra credit)
Spring 2011

Due Wednesday Feb. 16, 11AM (in class). No collaboration. Cite any and all sources. Please typeset your solutions.

1. Devise an algorithm that will add two regular redundant binary numbers of n and m digits in $O(\min\{n, m\})$ time worst-case. The output of your algorithm should be a regular redundant binary number, represented in the same way (same data structure) as the inputs (so that one could use the output as an input to a subsequent addition). Give a clear and concise description of your algorithm, in English or pseudocode or actual commented code. Verify that your algorithm is correct and that it has the desired time bound. Your algorithm is allowed to destroy the inputs in the process of building the output.

2. Consider a list containing the integers 1 through 10 in increasing order. Consider the access sequence 10, 8, 6, 4, 2, 9, 7, 5, 3, 1. If move-to-front (MTF) is used to do the accesses, what is the final list, after all the accesses? How many inversions does the final arrangement contain? (An inversion is a pair of list elements i and j such that $i < j$ but j precedes i in the list.)

3. Give an algorithm that counts the number of inversions in a fixed permutation of 1 through n. Your algorithm should run in $O(n\log n)$ time in the worst case. Give a clear and concise description of your algorithm, in English or pseudocode or actual commented code. Verify that your algorithm is correct and that it has the desired time bound. You may use known algorithms or data structures from COS 226, and their known running times.

4. Consider the following simplified model of self-adjusting lists. We restrict our attention to methods that access the k^{th} item from the front by traversing the list from the first item until reaching the desired item, and then optionally move the accessed item to any position nearer the front (without doing any other rearrangement). The cost to access the k^{th} item is k, no matter where it is moved. (It need not be moved at all.) Show that among such algorithms MTF is 2-competitive: on any access sequence, its total cost is no more than twice the cost of the minimum-cost off-line algorithm for that sequence.

5. Provide three links (url's) to interesting material discussing self-adjusting (also known as self-organizing) lists, competitive analysis, amortized efficiency, paging, and/or number systems that avoid carry propagation.

6. (Extra credit) We proved in class that if we use a regular or strictly regular redundant binary representation, adding 1 changes at most 3 digits, which means that counting from 0 to n takes at most $3n$ digit changes, whereas counting to from 0 to n using the standard binary representation takes at most $2n$ digit changes. But is the $3n$ tight? Use an amortized analysis to obtain the best upper bound you can (best constant factor) on the total number of digit changes to count from 0 to n using (a) the regular representation (regular addition) and (b) the strictly regular representation (strictly regular addition). This problem is extra credit because I don't (yet) know the answer.

Clarification: When adding 1 to a number whose last digit is a 2, such as 1012, we only charge one for changing the last digit, even though it first changes to 0 (fixing the rightmost 2 changes the number to 1020) and then changes to 1 (after adding 1, the number is 1021). Thus in this case the number of digit changes is two, not three. That is, we count digits changed per addition rather than digit changes per addition.

P.S. Now I know the answer to this problem. It is not so hard.