Final Exam

Algorithms and Their Complexity

February 10, 2000, 11am-14pm

Please do the exam on your own, with no help from books, notes or friends. Answer *three* of the following questions. You may answer either in Hebrew or in English. Please keep handwriting readable. Write your name and ID number on each page that you hand in. You may assume as known anything taught in class or given as homework.

1. Let $X = x_1, \ldots, x_n$ be a list of n elements (where n is odd). We have seen in class that *quicksort* sorts the list in expected $2n \ln n$ comparisons. Design a randomized algorithm that finds the median in $O(n)$ expected number of comparisons. Give both upper and lower bounds on the leading constant for the expected number of comparisons of your algorithm. In this exam it suffices that the bounds are reasonably close (e.g., within a factor of 2 of each other) rather than exact. (A remark that may be useful for the analysis of your algorithm: the integral of $1/x$ is $\ln x$ and the integral of $\ln x$ is $x \ln x - x$.)

2. We are given a set S of n elements and a list of m *triples*, where each triple contains three elements from S. A *3-dimensional matching* is a collection of mutually disjoint triples. In class we remarked that finding a 3-dimensional matching of maximum cardinality is NP-hard.

   (a) Show that the problem can be solved in time $O(n^c 2^n)$, for some constant $c > 0$. (Namely, the running time can be bounded as exponential in n rather than exponential in m.)

   (b) Give a (random) polynomial time algorithm for finding $\log n$ disjoint triples.

3. Given a complete undirected graph with edge lengths obeying the triangle inequality ($w(i,k) \leq w(i,j) + w(j,k)$), the *metric TSP* problem asks for a minimum length cyclic tour visiting each vertex exactly once. We have seen in class an algorithm that approximates metric TSP within a ratio of 3/2. A natural greedy algorithm for metric TSP creates partial solutions and extends them as follows:

   (a) Sort the edges in order of increasing length.

   (b) For each edge in order, add it to the solution unless it contradicts edges already in the solution. Namely, the new edge is not allowed to be incident with a vertex that already has degree 2, or to close a cycle of length less than n.

   Show that this algorithm approximates metric TSP within a factor of at most $H_n = \sum_{i=1}^{n} 1/i$.

4. A *proper 3-CNF formula* is the logical *and* of clauses, where each clause is the logical *or* of exactly 3 literals. For example:

$$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_4 \vee x_1 \vee x_3)$$

Finding a satisfying assignment for a proper 3-CNF formula is NP-hard. A *very proper 3-CNF formula* has the extra property that each variable appears exactly 3 times. Give a polynomial time algorithm for finding satisfying assignments for very proper 3-CNF formulas.

5. We have seen in class that two n-bit integers can be multiplied in time $O(n^{\log_2 3})$, rather than $O(n^2)$. Show a multiplication algorithm that works in $O(n(\log n)^c)$ time, for some constant $c > 0$. Here are some "hints" and assumptions to start you off (correct solutions not using these hints are also okay).

   (a) Observe that an n bit number, whose bit representation is $a_{n-1} \ldots a_1 a_0$ has value $\sum_{i=0}^{n-1} a_i x^i$ with $x = 2$.

   (b) You may assume that each of the two input numbers is at most $n/2$ bits long, and that n is an exact power of 2.

   (c) The fraction of prime numbers is large enough to make it likely that for some $k < n$, the number $p = kn + 1$ is prime. Assume that you are given such a prime p, and also a generator g for Z_p^*.

Good luck!