Data Structures - Assignment no. 2

Date due: 6/8/02

Remarks:

- If you fail to understand a question, or find a question erroneous, please contact me at `puzne@tau.ac.il`.

1. Prove the first statement of the Master Theorem:
   Let T be given recursively by $T(n) = aT(n/b) + f(n)$, where $a > 0$ and $b > 1$ are constants, and where $f \prec n^{\log_b a - \epsilon}$ for some positive constant ϵ. Then $T(n) \sim n^{\log_b a}$.

2. Prove or disprove each of the following statements, concerning natural-valued functions over the natural numbers.

   (a) If $f \prec h_1$ and $g \prec h_2$ then $f \cdot g \prec h_1 \cdot h_2$.

   (b) If $f \prec h_1$ and $g \prec h_2$ then $f^g \prec h_1^{h_2}$.

3. Determine the time complexity (order of growth) of the 2/3-sort algorithm shown in class.

4. Find the order of growth for the following recursively given functions. Solve the recurrences directly, that is without using the Master Theorem.

   (a) $T(n) = 2T(\frac{n}{2} + 14) + n$ (Hint: to prove that $T(n) \prec n \log n$, use the inductive hypothesis $T(k) \leq C(k \log k - 14 \log k)$).

   (b) $T(n) = T(cn) + T((1-c)n) + n$ where $0 < c < 1$ is constant.

   (c) $T(n) = \sqrt{n} T(\sqrt{n}) + n$.

5. Let $a_1, \ldots, a_k$ be positive real numbers satisfying $\sum_{i=1}^{k} a_i < 1$. Prove (formally) that a function T satisfying the recursion $T(n) = \sum_{i=1}^{k} T(a_i n) + n$, must be of order $\Theta(n)$.