Algorithms – CS-27200/37000   Homework – January 26, 2004
Instructor: László Babai   Ry-164   e-mail: `laci@cs.uchicago.edu`

ADVICE. Take advantage of the TA's office hours Monday, Tuesday and Thursday 5–6pm in the Theory lounge (Ry–162).

DATES TO REMEMBER. Wed Feb 4: Midterm 1; Mon Feb 23: Quiz 2.

HOMEWORK. Please **print your name on each sheet.** Print "U" next to your name if you seek 27200 credit and "G" if you seek 37000 credit. Please try to make your solutions readable. Unless expressly stated otherwise, all solutions are due at the **beginning of the next class.**
    Homework is collected in three separate piles (U, G, "G only"). Please write your solutions to graduate problems on **separate sheets.**

8.1 (U,G) **(6 points)** Draw the diagram of Batcher's odd-even MERGE network for merging two sorted lists of 8 items each (total 16 items). How many phases will this netwrok require? The two sorted lists should come on the first 8 wores and the last 8 wires, respectively. Make your drawing neat.

8.2 (U,G) **(6 points)** Let H be a heap and x the address of a node in H. Let us change the value $key(x)$ to $newkey(x)$ and suppose $newkey(x) < key(x)$. Write a PSEUDOCODE to restore the heap structure. Estimate the number of comparisons made in the process. Your estimate should be a function of n, the number of items in the heap (which is the number of nodes of the heap). Your estimate should be asymptotically best possible (with the correct constant).

8.3 (G only) **(10 points)** Suppose we are given n data (reals) arranged in a heap. Prove: sorting the data still requires asymptotically $n \log n$ comparisons. (The meaning is that "preprocessing" the data by arranging them in a heap does not significantly reduce the cost of sorting.) WARNING: this problem is NOT about the HEAPSORT algorithm. We are allowed to pick any two items, compare them, record the result, and choose our next step as a function of the result. The method does not need to have anything to do with the heap structure. Your lower bound must be valid for *all conceivable algorithms.*