Theory of Algorithms Final January 18, 2007

Write your name and a code of at least 5 characters, in print, onto each sheet you turn in. The code will be used to put up the results on the web page `http://www.cs.bme.hu/~pbiro/thalg.html` preserving your privacy.

1. Are the following statements true? Explain why or why not in each case.

   (a) $2^{n+1} = O(2^n)$

   (b) $2^{2n} = O(2^n)$

2. Banks often record transactions on an account in order of the times of the transactions, but many people like to receive their bank statements with checks listed in order by check number. People usually write check in order by check number, and merchants usually cash them with reasonable dispatch. The problem of converting time-of-transactions ordering to check-number ordering is therefore the problem of sorting almost-sorted input. Argue that the procedure INSERTION-SORT would tend beat the procedure QUICKSORT on this problem.

3. Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\lg n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y.

4. We have given an unsorted array of size n. Can we build a binary search tree from its elements in $O(n)$ time? Explain how or why not.

5. Consider inserting the keys 6, 12, 8, 22, 37, 10 into a hash table of lenght $m = 7$ using open addressing with the primary hash function $h'(k) = k \bmod m$. Illustrate the result of inserting these keys using linear probing, and using quadratic probing with $c_1 = 3$ and $c_2 = 1$.

6. Describe a red-black tree on n keys that realizes the largest possible ratio of red internal nodes. What is this ratio? What tree has the smallest possible ratio, and what is the ratio?

7. In our computer we work with n different types of picture-files, and we have r programs, each of them can transform one type to another one (for example, the program $p_{i,j}$ can transform a file with type t_i to an other file with type t_j). From the type t_i the *distance* of type t_j is the minimum number of transformations that is needed to convert a file with type t_i to a file with type t_j. (Note, that if we have a program $p_{i,j}$, then the distance from t_i to t_j is 1.) We would like to store our pictures using one single type t^*, in such a way that the maximum distance from t^* to any other type should be minimal. Give an $O(n(n+r))$-time algorithm that finds one of these best types.

Each problem is worth 10 points, you need 28 points at least to pass. Your point total in the final exam counts for 70% of your final grade.