Universität
des
Saarlandes

Kurt Mehlhorn, Konstantinos Panagiotou and Reto Spöhel

WS 2010-11

Models of Computation, an Algorithmic Perspective

Assignment 10 Tue 4.1.2011

This assignment is **due on January 12/14** in your respective tutorial groups. You are allowed (even encouraged) to discuss these problems with your fellow classmates. All submitted work, however, must be *written individually* without consulting someone else's solutions or any other source like the web.

Exercise 1 In the lecture we had the following strategy for implementing a stack in external memory. We keep a buffer of size B. When the buffer is full, we write it to disk, when the buffer is empty, we fill it from disk. Recall the argument from class that there are sequences of N stack operations that require $\Omega(N)$ IOs.

Now investigate the following randomized strategy for implementing a stack in external memory? Choose a random integer $i \in [0, B-1]$ and leave the first i positions in the first block unused. Then proceed as in class. Show that the expected number of I/Os required for any sequence of N operations is $O(N/B)$.

Exercise 2 A large undirected graph is given as a sequence of pairs of node IDs. Determine the number of triangles in the graph. Use only scans and sorts. A good start is to scan the sequence and to insert for each pair (i, j), the pair (j, i) into the sequence. Then you sort the resulting sequence. Then ...

Our solution requires space $O(\sum_v d_v^2)$ where d_v is the degree of v. How much space does your solution need? What is the maximum/minimum/average value of $\sum_v d_v^2$ for a graph with n nodes and m edges?

Exercise 3 Explore sorting by distribution. If the number N of elements is less than M, use an optimal internal sorting method. Otherwise, choose a random sample of size approximately M/B, e.g., by choosing elements independently with probability $M/(NB)$. Sort the sample internally and then distribute the input into buckets as defined by the sample. Allot a buffer of size B for each bucket. Sort each bucket recursively.

Assume first that the split into buckets is perfect, i.e., each bucket has size NB/M. What is the I/O-complexity of the method? Then discuss deviations from perfection.