Universität
des
Saarlandes

**Kurt Mehlhorn, Konstantinos Panagiotou
and Reto Spöhel**

WS 2010-11

Models of Computation, an Algorithmic Perspective

Assignment 11 Tue 11.1.2011

This assignment is **due on January 19/21** 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 We study matrix multiplication.

The standard way to store matrices is row-major order, i.e., a matrix A of size n by n is stored as the sequence $A_{1,1}, \ldots A_{1,n}, A_{2,1}, \ldots, A_{2,n}, \ldots$.

How many I/Os are required for matrix multiplication, when matrices are stored in row-major order?

How many I/Os are required for a matrix product AB, when A is stored in row-major order and B is stored in column-major order?

Exercise 2 Show that one can transpose a N by N matrix with $sort(N^2)$ IOs. Show that one can transform row-major layout into column-major layout with $sort(N^2)$ IOs.

Exercise 3 Block Layout of Matrices) Divide A into square submatrices of size $\sqrt{M} \times \sqrt{M}$. Store the elements in each submatrix contiguously

Visualize the layout for a 8 by 8 matrix that is split into submatrices of size 2 by 2, i.e., draw a 8 by 8 grid and number the grid cells in the order in which they appear in the layout.

What is the IO-complexity of matrix multiplication when matrices are stored in block layout?

How does this compare to the bound obtained in the first exercise?

Exercise 4 (Recursive Layout of Matrices) Split a matrix A into four matrices $A_{1,1}$, $A_{1,2}$, $A_{2,1}$, $A_{2,2}$. Store A as $A_{1,1}$ followed by $A_{1,2}$ followed by $\ldots$. Use the same technique recursively on the submatrices.

Visualize the layout for a 8 by 8 matrix.

What is the IO-complexity of matrix multiplication when matrices are stored in block layout?

Does the algorithm need to know M?

How does this compare to the bounds obtained in the two preceding exercises?

Exercise 5 Redo the preceding exercise for Strassen matrix multiplication.

Exercise 6 Work through the following lower bound proof.

A Lower Bound We prove a lower bound of $\Omega(\frac{n}{M^{1/2}B})$ IOs under the following assumption. Each array element is associated to a storage location. For each triple (i, j, k), $1 \le i, j, k \le n$, it is necessary that $A_{i,j}$ and $B_{j,k}$ and $C_{i,k}$ are *simultaneously* in main memory.

Observe that the model is quite restrictive, e.g., Strassen matrix multiplication is not covered by the model.

We start with a combinatorial lemma.

Lemma 1. *An undirected graph with m edges contains at most $4m^{3/2}$ triangles, i.e., triples (u, v, w) of distinct vertices such that uv, vw and uw are edges of G.*

Proof. For each triangle T, we will distribute a charge of 1 over the edges of G. Then we will upper bound the total charge.

We identify the vertices with the integers 1 to n. For each triangle t, let $u(T)$ be the smallest vertex of the triangle.

Consider a fixed vertex u and consider the triangles T with $u = u(T)$. If the degree of u is at most $\sqrt{m}$, we charge $1/2$ to each edge of T incident to u. If the degree of u is more that $\sqrt{m}$, we charge $1/\sqrt{m}$ to each edge incident to u.

Consider now an edge $e = vw$. How much is charged to it? We estimate the charge due to endpoint v. If the degree of v is at most $\sqrt{m}$ then e picks up no charge of the second kind and at most $\sqrt{m}/2$ charge of the first kind, because there are at most $\sqrt{m}$ choices for the other edge in the triangle incident to v. If the degree of v is more than $\sqrt{m}$ then e picks up no charge of the first kind and at most $m/\sqrt{m}$ charge of the second kind because there are at most m choices for the triangle edge not incident to v. Thus the total charge to e because of endpoint v is at most $2\sqrt{m}$.

The total charge to all edges is at most $4m^{3/2}$. □

Theorem 1. *In the model defined above, the IO-complexity of matrix multiplication is at least $\Omega(n^3/(M^{1/2}B)$.*

Proof. We partition execution into slices of M/B IOs. In a slice, we can read/write at most M array elements. Thus at most $2M$ array elements "meet" during a slice, namely the M elements that were in memory at the beginning of the slide plus the up to M elements that were read in the slice. The $2M$ array elements can form at most $4 \cdot (2M)^{3/2}$ triangles. Since we need to form n^3 triangles, there must be at least

$$\Omega\left(\frac{n^3}{M^{3/2}}\right)$$

slices. $\square$

Exercise 7 Develop an external memory algorithm to find connected components of a graph. Start from the PRAM-algorithm presented in class.