Prof. Friedrich Eisenbrand
Gennady Shmonin

Due Date: October 20, 2008

Combinatorial Optimization

Fall 2008

Assignment Sheet 2

Exercise 1 (Maximal matching)
Let $G = (V, E)$ be a graph. A matching M in G is a *maximal matching* if it is not a proper subset of another matching in G. M is a *maximum matching* if it has the maximum cardinality among all matchings in G. Prove that the cardinality of any maximal matching is at least half the size of a maximum matching.

Exercise 2 (Matrix multiplication)
Describe an algorithm which, provided two integral $n \times n$ matrices A and B, computes the matrix $C = AB$ using $O(n^{\log_2 7})$ multiplications.

Hint. Assuming that n is even, split A, B and C into equal-size blocks:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \qquad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}.$$

Define matrices

$$M_1 = (A_{11} + A_{22})(B_{11} + B_{22}), \qquad M_5 = (A_{11} + A_{12})B_{22},$$
$$M_2 = (A_{21} + A_{22})B_{11}, \qquad M_6 = (A_{21} - A_{11})(B_{11} + B_{12}),$$
$$M_3 = A_{11}(B_{12} - B_{22}), \qquad M_7 = (A_{12} - A_{22})(B_{21} + B_{22})$$
$$M_4 = A_{22}(B_{21} - B_{11}),$$

and prove that

$$C_{11} = M_1 + M_4 - M_5 + M_7, \qquad C_{21} = M_2 + M_4$$
$$C_{12} = M_3 + M_5, \qquad C_{22} = M_1 - M_2 + M_3 + M_6.$$

Exercise 3 (Expected running time)
Let $N = \{1, 2, \ldots, n\}$. For some purposes, a programmer needs to choose at random m pairwise different numbers from N. He comes up with the following procedure:

$M := \emptyset$; $i := 1$
while $i \le m$ **do**
    pick $n \in N$ uniformly at random

if $n \notin M$ **then** $\{ M := M \cup \{n\};\ i := i + 1 \}$
endwhile

The algorithm indeed produces a randomly chosen set of m different numbers from N. What is the expected running time of this algorithm?

Exercise 4 (k-median)
Consider the following algorithm to compute the k-median (the k-th smallest element) of a set S of n different numbers.

MEDIAN(S, k)
choose y from S uniformly at random
$S_1 := \{x \in S : x < y\}$, $S_2 := \{x \in S : x > y\}$
if $|S_1| \geq k$ **then return** MEDIAN(S_1, k)
if $|S_1| = k - 1$ **then return** y
if $|S_1| < k - 1$ **then return** MEDIAN(S_2, $k - |S_1| - 1$)

Show that the expected running time of the algorithm is $O(n)$.

Exercise 5 (FastCut)
Consider the following algorithm for finding a minimum cut in a graph.

FASTCUT(G, n)
if $n > 6$ **then**
$t := \lceil \frac{n}{\sqrt{2}} + 1 \rceil$
perform $n - t$ random edge contractions in G to obtain G'
$z' :=$ FASTCUT(G', t)
perform $n - t$ random edge contractions in G to obtain G''
$z'' :=$ FASTCUT(G'', t)
return MIN(z', z'')
else
use brute-force algorithm to compute min-cut
endif

Show that the running time of this algorithm is $O(n^2 \log n)$ and the probability that the algorithm returns a minimum cut is $\Omega(1/\log n)$.