

IMPORTANT: Write each problem on a separate sheet of paper. Write your name on each sheet. (Why? To speed up the grading we will use parallelization—each problem will be graded by a different TA. At the beginning of the class (on the date when the homework is due) there will be a separate pile for each problem.)

Homework 5 is due: Dec. 3 (Thursday); collected before class in B&L 109.

Problem sessions before Homework 4 is due:

Nov. 30 (Monday) 6:30pm - 7:30pm in Goergen 108

Dec. 1 (Tuesday) 6:00pm - 7:00pm in Goergen 108

Dec. 2 (Wednesday) 6:15pm - 7:15pm in Goergen 108

Dec. 2 (Wednesday) 7:30pm - 8:30pm in Goergen 108

Exam 4 is on: Dec. 10 (Thursday).

Problem sessions before Exam 3:

Dec. 7 (Monday) 6:30pm - 7:30pm in Goergen 108

Dec. 8 (Tuesday) 6:00pm - 7:00pm in Goergen 108

Dec. 8 (Wednesday) 6:15pm - 7:15pm in Goergen 108

Dec. 9 (Wednesday) 7:30pm - 8:30pm in Goergen 108

Comprehensive Final Exam is on Dec. 19 (Saturday) 4:00pm - 7:00pm in B&L 109.

1 Homework - solve and turn in

5.1 (due Dec 3) We are given a string S over alphabet $\Sigma = \{a, b\}$. We want to find a cyclic rotation of S that disagrees with the original string S on the maximal number of places.

For example, if $S = ababb$ then there are 5 cyclic rotations of S : $ababb$ (0), $babba$ (4), $abbab$ (2), $bbaba$ (2), $babab$ (4); the numbers in the parenthesis give the number of places on which the rotations disagree with S . Thus the output should be either $babba$, or $babab$.

Give an $O(n \log n)$ -time algorithm for the problem.

5.2 (due Dec 3) Let A and B be two sets. Their sum $A + B$ is defined to be

$$A + B := \{a + b \mid a \in A, b \in B\},$$

for example if $A = \{1, 4, 6\}$ and $B = \{2, 5\}$ then $A + B = \{3, 6, 8, 9, 11\}$. Give $O(n \log n)$ algorithm whose input is two sets $A, B \subseteq \{1, \dots, n\}$ and the output is set $A + B$.

5.3 (due Dec 3) The Hadamard matrices H_0, H_1, \dots , are defined as follows.

- H_0 is the 1×1 matrix $[1]$.
- For $k > 0$, H_k is the $2^k \times 2^k$ matrix

$$H_k = \left[\begin{array}{c|c} H_{k-1} & H_{k-1} \\ \hline H_{k-1} & -H_{k-1} \end{array} \right].$$

Show that if v is a column vector of length $n = 2^k$ then the matrix-vector product $H_k v$ can be calculated using $O(n \log n)$ operations. (Assume that all the numbers involved are small enough that basic arithmetic operations take $O(1)$ time.)

2 Bonus Homework - solve and turn in

5.4 (due Dec 3) The *max-weight* of a spanning tree T is the maximum weight of an edge of T . A *min-max-weight spanning tree* is a spanning tree with the minimum max-weight. Give $O(E+V)$ algorithm which finds min-max-weight spanning tree of a given input graph G .