Divide&Conquer, FFT, NP, ILP (part 4 of CSC 282),

http://www.cs.rochester.edu/~stefanko/Teaching/19CS282

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. 6 (Thursday); collected before class in Wegmans 1400.

Problem sessions before Homework 5 is due:
Dec. 2 (Monday) 6:15pm - 7:15pm in Hylan 202
Dec. 3 (Tuesday) 6:15pm - 7:15pm in in Goergen 109
Dec. 4 (Wednesday) 6:15pm - 7:15pm in Gavett 310
Dec. 4 (Wednesday) 7:40pm - 8:40pm in Hutchinson 473

Comprehensive Final Exam is on Dec. 19 (Thursday) 4:00pm - 7:00pm in Douglass Ballroom.

1 CSC 482 Homework—solve and turn in (bonus for CSC 282)

5.1 (**due Dec 5**) The Hadamard matrices $H_0, H_1, \ldots,$ 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 CSC 282/482 Homework—solve and turn in

Let $a_0, a_1 \ldots, a_{n-1}$ and $b_0, b_1 \ldots, b_{n-1}$ be two sequences of integers. The **convolution** of the two sequences is a sequence $c_0, c_1, \ldots, c_{2n-2}$ where

$$c_k = \sum_i a_i b_{k-i}.$$

Thus, for example $c_0 = a_0 b_0, c_1 = a_0 b_1 + a_1 b_0, \ldots, c_{2n-2} = a_{n-1} b_{n-1}$. You will need the following fact for the first two problems (5.2 and 5.3): the convolution of two sequences can be computed in time $O(n \log n)$.

5.2 (**due Dec 5**) Let A and B be two sets. Their sum $A + B$ is defined to be

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

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 $A, B \subseteq \{1, \ldots, n\}$ and the output is set $A + B$. (Hint: create two sequences of length n whose convolution will allow you to compute the set $A + B$).

5.3 (**due Dec 5**) 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.

5.4 (**due Dec 5**) We are given an undirected graph $G = (V, E)$ and for each edge $e = \{u, v\} \in E$ we are given two numbers a_e, b_e. We want to assign a number from $\{1, \ldots, 10\}$ to each vertex in V so that for every edge $e = \{u, v\} \in E$ the sum of the numbers assigned to u and v is either a_e or b_e. Does such an assignment exist? Write an integer linear program for the problem. Clearly describe the variables in your integer linear program.

Problem 5.5 (see the next page) is part of the CSC 282/482 homework.

5.5 (**due Dec 5**) **Print this page and complete this problem on the printed page.** For each question below, circle one of the three answers and add a short explanation in the space immediately below that question. The problem POLYNOMIAL EQUATIONS OVER INTEGERS (from "Unsolvable problems" box in the DPV textbook) is: given polynomial equation in many variables, does it have an integer solution? Recall that a problem Q is **NP-hard** if 3-SAT reduces to Q. A problem Q is **NP-complete** if Q is **NP-hard** and Q is in **NP**.

1. Every problem in **NP** is in **P**. TRUE - FALSE - WE DON'T KNOW

2. There exists a problem in **NP** which is in **P**. TRUE - FALSE - WE DON'T KNOW

3. There exists a problem in **NP** which is not in **P**. TRUE - FALSE - WE DON'T KNOW

4. There exists an **NP-complete** problem which is in **P**. TRUE - FALSE - WE DON'T KNOW

5. Every **NP-complete** problem is in **P**. TRUE - FALSE - WE DON'T KNOW

6. POLYNOMIAL EQUATIONS OVER INTEGERS is in **P**. TRUE - FALSE - WE DON'T KNOW

7. POLYNOMIAL EQUATIONS OVER INTEGERS is in **NP**. TRUE - FALSE - WE DON'T KNOW

8. POLYNOMIAL EQUATIONS OVER INTEGERS is **NP-hard**. TRUE - FALSE - WE DON'T KNOW

9. Every **NP-hard** problem is in **P**. TRUE - FALSE - WE DON'T KNOW

10. FACTORING is in **P**. TRUE - FALSE - WE DON'T KNOW

11. If **P** = **NP** then FACTORING is in **P**. TRUE - FALSE - WE DON'T KNOW

12. If **P** = **NP** then every **NP-complete** problem is in **P**. TRUE - FALSE - WE DON'T KNOW

13. If **P** = **NP** then some **NP-complete** problem is in **P**. TRUE - FALSE - WE DON'T KNOW

14. If some **NP-complete** problem is in **P** then **P** = **NP**. TRUE - FALSE - WE DON'T KNOW

15. Every problem in **P** is in **NP**. TRUE - FALSE - WE DON'T KNOW