CSE 565	Martin Fürer	Algorithm Design and Analysis

Assignment 9 Given: 11/8/18 Due: Wednesday 11/14/18

Exercises

Exercises are for your own practice. Don't hand them in. A version of them could show up in an exam though.

1. Solve Exercise 1, Chapter 7 on page 415 of the Textbook.
   (Min cuts)

2. Solve Exercise 5, Chapter 7 on page 416 of the Textbook.
   (Add 1 to each capacity. Does a mincut remain?)

Problems

Problem solutions have to be handed in. A subset of them will be graded.

1. [5+5=10 points] Solve Problem 2, Chapter 7 on page 415 of the Textbook.
   (Flow and cut example)

2. [20 points] Solve Problem 11, Chapter 7 on page 420 of the Textbook.
   (How bad is a greedy flow?)

3. [20 points] Solve Problem 12, Chapter 7 on page 420 of the Textbook.
   (Algorithm to delete k edges) You may call a known flow algorithm as a procedure without describing it.

4. [$5 \times 5 = 25$ points] You are given $n = 2^k$ polynomials

   $$P_i(x) = \sum_{j=0}^{n-1} a_{ij} x^j \text{ for } 0 \leq i < n$$

   of degree $n-1$ each. The complex coefficients a_{ij} are represented by pairs of floating point numbers. You are also given a table with $1, \omega, \omega^1, \ldots, \omega^{n-1}$, where ω is the primitive n-th root of unity $e^{2\pi i/n}$. Now assume, you want to compute the product of these n polynomials $P_0 \cdot P_1 \cdot \ldots \cdot P_{n-1}$. We count each arithmetic operation on real numbers as 1 step, even though one should actually worry whether multi-precision arithmetic is needed for large n.

For each answer give just enough justification, to show how you computed your answer.

(a) Assume, you compute the product from left to right

$$(\ldots((P_0 \cdot P_1) \cdot P_2) \cdot \ldots \cdot P_{n-1})$$

with school multiplication (brute force). What is the asymptotic running time?

(b) Assume, you compute the product $P_0 \cdot P_1 \cdot \ldots \cdot P_{n-1}$ in a more clever balanced way. How? (Just tell which product is computed last.)
What is the asymptotic running time with school multiplication (brute force)?

(c) Assume, you use the same organization as in (b), but you employ FFT (Fast Fourier Transform) for each multiplication. What is the asymptotic running time now?

(d) In (c), you do every multiplication with the help of 2 FFT's and 1 inverse FFT. You notice that all but the last inverse FFT is immediately followed by an FFT. Why does this make sense? (One might think, they just cancel each other.)

(e) Now try to compute the same product by just n FFT's and 1 inverse FFT. The computation in between consists entirely of multiplications of values. What is the asymptotic running time now?