

## Homework 1

Due date: October 3, 2013

---

1. (a) For each of the following, indicate whether  $f(n) = O(g(n))$ ,  $f(n) = \Omega(g(n))$ , or  $f(n) = \Theta(g(n))$ . Justify your answer.

- $f(n) = (\log \log n)^5$  and  $g(n) = \frac{\log n}{\log \log n}$ ;
- $f(n) = (\log n)^{\log_2 n}$  and  $g(n) = \frac{n}{\log n}$ ;
- $f(n) = (\log n)^{\log_2 n}$  and  $g(n) = 2^{(\log_2 n)^{1.5}}$ .

- (b) Time complexity of some algorithm is given by a recurrent relation

$$T(n) = a \cdot T(n/2) + O(n^b) \quad \text{and} \quad T(1) = 1 ,$$

where  $a > 0$ ,  $b \geq 0$ . Find a closed form expression for  $T(n)$ .

Hint. Consider three cases:  $b > \log_2 a$ ;  $b = \log_2 a$  and  $b < \log_2 a$ .

2. By using Fast Fourier Transform (FFT) algorithm, evaluate the polynomial  $A(x) = 3x^4 + 2x^3 - 2x^2 - x$  at the complex 6th roots of unity. Show at least two levels of recursion.
3. Let  $A(x) = x^2 + x + 2$  and  $B(x) = 2x + 3$ . In this question, we will compute the polynomial  $C(x) = A(x) \cdot B(x)$  by using the FFT algorithm.

- (a) What is the minimum number of points we need to use? Explain.
- (b) Evaluate  $A(x)$  at the complex 4th roots of unity. Show at least two levels of recursion.
- (c) Evaluate  $B(x)$  at the complex 4th roots of unity. Show at least two levels of recursion.
- (d) Compute  $C(x)$  at the complex 4th roots of unity.
- (e) Find the coefficients of  $C(x)$ .

4. Let  $a$  and  $b$  be two  $n$ -bit nonnegative integers, and let

$$(a_{n-1}, a_{n-2}, \dots, a_1, a_0) \quad \text{and} \quad (b_{n-1}, b_{n-2}, \dots, b_1, b_0)$$

be their respective binary representations. Define integer  $c = a \cdot b$ . In this question we are interested in algorithm for computing  $c$ .

- (a) Define polynomials

$$A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$$

and

$$B(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0 .$$

Show that  $a = A(2)$  and  $b = B(2)$ .

- (b) Define polynomial  $C(x) = A(x) \cdot B(x)$ . (Note that the polynomial  $C(x)$  can be computed from  $A(x)$  and  $B(x)$  using FFT algorithm.) Is  $C(2) = c$ ? Prove or disprove.
- (c) What is the degree  $m$  of  $C(x)$ ?
- (d) Observe that the coefficients of  $C(x)$  are not necessarily all 0 and 1. Propose an algorithm that computes the binary representation of  $c$ , namely

$$(c_m, c_{m-1}, \dots, c_1, c_0),$$

from the coefficients of  $C(x)$  using only  $O(n)$  arithmetic operations over integers with  $O(\log n)$  bits.