Neatly print your name(s), NetID(s), and the alias(es) you used for Homework 0 in the boxes above. Staple this sheet to the top of your homework. If you are on campus, submit the homework by submitting it in SC 3306 (or sliding it under the door).

**Note:** You will be held accountable for the appropriate responses for answers (e.g. give models, proofs, analysis, etc). For NP-Complete problems you should prove everything rigorously, i.e. for showing that it is in NP, give a description of a certificate and a polynomial time algorithm to verify it, and for showing problems are NP-Hard, you must show that your reduction is polynomial time (by similarly proving something about the algorithm that does the transformation) and proving both directions of the ‘if and only if’ (a solution of one is a solution of the other) of the many-one reduction.
What do you know? You know just what you perceive.
What can you show? Nothin’ of what you believe,
And as you grow, each thread of life that you leave
Will spin around your deeds and dictate your needs
As you sell your soul and you sow your seeds,
And you wound yourself and your loved one bleeds,
And your habits grow, and your conscience feeds
On all that you thought you should be –
I never thought this could happen to Meeeee
– Dreidel, Don McLean

Required Problems

1. **3SUM**
   **[20 Points]**
   Consider two sets $A$ and $B$, each having $n$ integers in the range from 0 to $10n$. We wish to compute the Cartesian sum of $A$ and $B$, defined by
   
   $$ C = \{ x + y : x \in A \text{ and } y \in B \}. $$
   
   Note that the integers in $C$ are in the range from 0 to $20n$. We want to find the elements of $C$ and the number of times each element of $C$ is realized as a sum of elements in $A$ and $B$. Show that the problem can be solved in $O(n \log n)$ time. (Hint: Represent $A$ and $B$ as polynomials of degree at most $10n$.)

2. **COMMON SUBSEQUENCE**
   **[10 Points]**
   Given two sequences, $a_1, \ldots, a_n$ and $b_1, \ldots, b_m$ of real numbers, We want to determine whether there is an $i \geq 0$, such that $b_1 = a_{i+1}, b_2 = a_{i+2}, \ldots, b_m = a_{i+m}$. Show how to solve this problem in $O(n \log n)$ time with high probability using FFT.

3. **Computing Polynomials Quickly**
   **[20 Points]**
   In the following, assume that given two polynomials $p'(q), q'(x)$ of degree at most $n$, one can compute the polynomial remainder of $p'(x)$ mod $q'(x)$ in $O(n \log n)$ time. The remainder of $r'(x) = p'(x) \mod q'(x)$ is the unique polynomial of degree smaller than this of $q'(x)$, such that $p'(x) = q'(x) \cdot d'(x) + r'(x)$, where $d'(x)$ is a polynomial.
   Let $p(x) = \sum_{i=0}^{n-1} a_i x^i$ be a given polynomial of degree $n - 1$.
   
   (a) **[4 Points]** Prove that $p(x) \mod (x - z) = p(z)$, for all $z$.
   (b) **[4 Points]** We want to evaluate $p(\cdot)$ on the points $x_0, x_1, \ldots, x_{n-1}$. Let
   
   $$ P_{ij}(x) = \prod_{k=i}^{j} (x - x_k) $$
   
   and
   
   $$ Q_{ij}(x) = p(x) \mod P_{ij}(x). $$
   
   Observe that the degree of $Q_{ij}$ is at most $j - i$.
   Prove that, for all $x$, $Q_{kk}(x) = p(x_k)$ and $Q_{0, n-1}(x) = p(x)$. 

2
(c) **[4 Points]** Prove that for $i \leq k \leq j$, we have

$$\forall x \quad Q_{ik}(x) = Q_{ij}(x) \mod P_{ik}(x)$$

and

$$\forall x \quad Q_{kj}(x) = Q_{ij}(x) \mod P_{kj}(x).$$

(d) **[8 Points]** Give an $O(n \log^2 n)$ time algorithm to evaluate $p(x_0), \ldots, p(x_{n-1})$. Here $x_0, \ldots, x_{n-1}$ are $n$ given real numbers.

4. **Matrix Madness**

**[20 Points]**

We can sort the entries of an $m \times m$ matrix by repeating the following procedure $k$ times:

1. Sort each odd-numbered row into monotonically increasing order.
2. Sort each even-numbered row into monotonically decreasing order.
3. Sort each column into monotonically increasing order.

(a) **[8 Points]** Suppose the matrix contains only 0’s and 1’s. We repeat the above procedure again and again until no changes occur. In what order should we read the matrix to obtain the sorted output ($m \times m$ numbers in increasing order)? Prove that any $m \times m$ matrix of 0’s and 1’s will be finally sorted.

(b) **[8 Points]** Prove that by repeating the above procedure, any matrix of real numbers can be sorted. [Hint: Refer to the proof of the zero-one principle.]

(c) **[4 Points]** Suppose $k$ iterations are required for this procedure to sort the $m \times m$ numbers. Give an upper bound for $k$. The tighter your upper bound the better (prove you bound).

5. **Lower Bound on Sorting Network**

**[10 Points]**

Prove that an $n$-input sorting network must contain at least one comparator between the $i$th and $(i + 1)$st lines for all $i = 1, 2, \ldots, n - 1$.

6. **First Sort, Then Partition**

**[10 Points]**

Suppose that we have $2n$ elements $< a_1, a_2, \ldots, a_{2n} >$ and wish to partition them into the $n$ smallest and the $n$ largest. Prove that we can do this in constant additional depth after separately sorting $< a_1, a_2, \ldots, a_n >$ and $< a_{n+1}, a_{n+2}, \ldots, a_{2n} >$. 

3