Problem Set 4

Due dates: Electronic submission of the pdf file of this homework is due on **10/3/2021 before 11.59pm** on canvas.

Name: [redacted]

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework. The solutions given in this homework are my own work.

Signature: [redacted]

Make sure that you describe all solutions in your own words. Typeset your solutions in LaTeX. Read chapters 30 and 16 in our textbook.

Problem 1. (20 points) Let ω be a primitive nth root of unity. The fast Fourier transform implements the multiplication with the matrix

$$F = (\omega^{ij})_{i,j\in[0..n-1]}.$$

Show that the inverse of the F is given by

$$F^{-1} = \frac{1}{n}(\omega^{-jk})_{j,k\in[0..n-1]}$$

[Hint: $x^n - 1 = (x-1)(x^{n-1} + \cdots + x + 1)$, so any power $\omega^\ell \neq 1$ must be a root of $x^{n-1} + \cdots + x + 1$.] Thus, the inverse FFT, called IFFT, is nothing but the FFT using ω^{-1} instead of ω, and multiplying the result with $1/n$.

Solution. For problem 1, let w be a primitive n^{th} root of unity. The FFT implements the multiplication with the matrix $F = (\omega^{ij})_{i,j\in[0..n-1]}$.Here, we have $x^n - 1 = (x-1)(x^{n-1} + ... + x + 1)$ so any power $w^i \neq 1$ must be the root of $(x^{n-1} + ... + x + 1)$. An element $w \in R$ is the n^{th} root of unity if $w^n = 1$. Its primitive when $w^i] \neq 1$ for all $1 \leq i \leq n-1$, if $w^d \neq 1$ for all divisors of n. In order to write FFT as a matrix will give us $M(w)_{ij} = w_{ij}$ for $0 \leq i, j \leq n-1$. So then we would get A= M(w) and B = $\frac{1}{n}M[w^{-1}]$ where the inverse matrices $(AB)_i = \frac{1}{n}\sum_{k=0}^{n-1} w^k w^{-jk} = \frac{1}{n}\sum_{k=0}^{n-1}(w^l)^k$, where $l = i - j \neq 0$ and take mod n where by multiplying that quantity by $1-w^1$ results in $\frac{1}{n}[1-w^n] = \frac{1}{n}[1-l] = 0$, hence we can show that $1-w^l$ isn't a 0 divisor and further prove that the inverse of F is $F^{-1} = \frac{1}{n}(\omega^{-jk})_{j,k\in[0..n-1]}$.

Problem 2. (20 points) Describe in your own words how to do a polynomial multiplication using the FFT and IFFT for polynomials $A(x)$ and $B(x)$ of degree $\leq n-1$. Make sure that you describe the length of the FFT and IFFT needed for this task. Be concise and precise.

Solution. For problem 2, we must describe how to do polynomial multiplication using FFT and IFFT. So given A(x) = $a_0 + a_1x + ... + b_{n-2}x^{n-2}$ and B(x)= $b_0 + b_1x + ... + b_{n-2}x^{n-2}$. This shows that for polynomials of A(x) and B(x) with degree $\leq n-1$, we can add n-1 to the higher order coefficients to A(x) and B(x), as shown in the equations above. In order to evaluate the two polynomials by using FFT, there must me 2n-2 points where we can use point wise multiplications for the point value forms. The product of the polynomials A(x) and B(x) by using the FFT will give us C(x) = $c_0 + c_1x + ... + c_{2n-4}x^{2n-4}$. The time used to find the product C(x) with the degree of n-1 using IFFT and FFT when the input/ output is in coefficient form is $\Theta((n-1)\log(n-1))$, and the length of the process is determined based on the time consumed.

Problem 3. (20 points) How can you modify the polynomial multiplication algorithm based on FFT and IFFT to do multiplication of long integers in base 10? Make sure that you take care of carries in a proper way. Write your algorithm in pseudocode and give a brief explanation.

Solution. For problem 3, we need to build an algorithm for multiplication of two large integers of base 10 using FFT. So let's say that A(x) and B(x) are two large integers of size n where the base is 10 which can also be referred to power of 10. So for the pseudocode, let A and B in the polynomial form by decomposing them in base x, where x is equal to 10. From the previous problem, we know that A(x) = $a_0 + a_1x + ... + b_{n-2}x^{n-2}$ and B(x)= $b_0 + b_1x + ... + b_{n-2}x^{n-2}$. So for the product, the degree of C is less than $a \leq n - 1$.
1) Based on the polynomial equation, we can input n number of points for $x_0, x_1, ..., x_{n-1}$
2) We can include A, so it contains the points $A(x_0), A(x_1), .., A(x_{n-1})$
3) Based on the polynomial equation, we can input n number of points for $x_0, x_1, ..., x_{n-1}$ again for B
4) Evaluate points of B, where $B(x_0), B(x_1), .., B(x_{n-1})$
5) By multiplying A(x) and B(x), determine $C(x_0), C(x_1), .., C(x_{n-1})$
6) Return the coefficients found in C by using interpolation.
This gives us the time complexity of $O(n \log n)$.

Problem 4. (20 points) Let $(S, \mathcal{F})$ be a matroid. A set B in the family $\mathcal{F}$ is called a **basis** if and only if it is not properly contained in any other set in the family $\mathcal{F}$. Show that if B and B' are two bases in $\mathcal{F}$, then $|B| = |B'|$.

[Write the solution in your own words. Be concise and precise. Do not give spoilers.]

Solution. For problem 4, we must show that $|B| = |B'|$. Given that $(S, \mathcal{F})$ is a matroid and in $\mathcal{F}$ A set B is called a basis if and only it isn't properly contained in another set in family $\mathcal{F}$. So given the properties of basis of a matroid, which is no base can properly contain another base, and if B and B' are the bases and e is element of B, then there's any element of B', where $(B - e \cup f)$ is a base. So given the two bases for matroid F, which are B and B', which each contain a different amount of elements, so $|B| < |B'|$. So if there is some element $e_1 \in S$, so $e_1 \in B$ but $e_1 \notin B'$. So if one were to remove that element from B, then by using the second property then some element $e_2 \in B'$, but then $e_2 \notin B$. We can say B" $= B-(e_1 \cup e_2)$ such that B" is in M and $|B| = |B"|$ and $|B'| \neq |B| = |B"|$. Repeating the same process k times will leave us with no element in thats not in base B_k, so $e \in B_k$ where e is also in B' and $B_k \subseteq B'$. However using the first property we know that no base can properly contain another, giving us a contradictory statement, so $B_k = B'$, hence saying that elements in base B are same as elements in base B', proving $|B| = |B'|$.

Problem 5. (20 points) Solve exercise 16.4-4 on page 443 of [CLRS].

Solution. For problem 16.4-4, where we must prove that the set of all sets A contain at most one member of each subset in the partition can determine the independent sets of a matroid. So let S be a finite set, and $S_1, S_2, ..., S_k$ be a partition of S where they would be nonempty disjoint sets. We know that, based on the condition given, $I = A : |A \cap S_i| \leq 1$ for i = 1,2,...k. Based on this, to show that (S, I) is a matroid, Let $x \subset y$ and $y \in I$, so then $(x \cap S_i) \subset (y \cap S_i)$,

for every i, so $|x \cap S_i| \leq |y \cap S_i| \leq 1$, where $1 \leq i \leq k$, and M is closed under inclusion. Next, let $A, B \in I$ with $|A| = |B| + 1$, then there must be some j such that $|A \cap S_j| = 1$, and $|B \cap S_j| = 0$. Now let a $= A \cap S_j$ so then $a \neq b$, and $|(B \cup a) \cap S_j| = 1$. We can say that, because $|(B \cup a) \cap S_i| = |B \cap S_i|$. We can now conclude that $i \neq j$, so $B \cup a \in I$, thus M is a matroid.

Checklist:

☐ Did you add your name?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
☐ Did you sign that you followed the Aggie honor code?
☐ Did you solve all problems?
☐ Did you submit the pdf file of your homework?