1. (25 pts) Applications of FFT

This problem covers two other instances where fast convolution using FFT can be used to speed up algorithms.

(a) (15 points) Given an integer $n$, count the number of Pythagorean triples $a, b, c \leq n$ such that $a^2 + b^2 \equiv c^2 \mod n$. Your algorithm should work in $O(n \log n)$ time.

**Bonus Problem:** Give a combinatorial algorithm for this problem that runs in $o(n)$ time.

(b) (20 points) Given a graph $G = (V, E)$ that is a tree with $n$ nodes rooted at $r$, along with a constant $k$, count the number of subsets of the vertices $S \subseteq V$ that satisfies the following conditions:

i. $r \in S$

ii. $|S| = k$,

iii. $S$ is connected

In other words, $S$ forms a subtree containing the root whose size is $k$. You might want to first provide a dynamic programming algorithm that runs in $O(n^3)$ time. Any solution that’s $O(n^2 \log n)$ or better will receive full points. You may ignore issues related to precision (assume we want the total number of subsets mod $p$ where $p$ is some small prime if you want).

**Solution:**
(a) Let $p(x)$ be a polynomial with the coefficient of $x^k$ being the number of integers $1 \leq j \leq n$ such that $j^2 \equiv k \mod n$. This polynomial has degree at most $n - 1$ and takes time $O(n)$ time to construct. Now consider $p(x)^2$. This polynomial has the coefficient of $x^k$ being the number of pairs of integers $1 \leq i, j \leq n$ such that $i^2 + j^2 \equiv k \mod n$. This polynomial has degree at most $2n - 2$. We fold the exponents of this polynomial mod $n$; so, let $p'(x)$ such that $x^k$ has coefficient equal to $k$th coefficient of $p(x)^2$ plus coefficient of $n + k$. Now multiply $p'(x)$ coefficient-wise by $p(x)$. This will give coefficient of $x^k$ being the number of triples $1 \leq i, j, \ell \leq n$ such that $i^2 + j^2 = \ell^2 \equiv k \mod n$. Summing over all $k$ gives the desired bound.

Multiplying two polynomials is $O(n \log n)$ by FFT, folding coefficients is $O(n)$, and multiplying coefficient-wise is $O(n)$. So, the overall runtime is $O(n \log n)$ as desired.

(b) We proceed somewhat recursively, essentially defining polynomials for every tree structure and multiplying them. For a leaf $\ell$, define the polynomial $p_\ell(x) = 1 + x$. What this encodes is that there is one way to choose 0 nodes to be in the tree, and one way to choose 1 nodes to be in the tree.

Now let $T$ be a tree with root $r$ and children $c_1, c_2, \ldots, c_m$. Let $p_1, p_2, \ldots, p_m$ denote the respective polynomials for the children. Then I claim $p_r = 1 + x \prod_{i=1}^{m} p_i$. Indeed, this simply follows for induction with base case being the leaves. The coefficient of $x^k$, $k \geq 1$ in $p_r$ corresponds exactly to the problem condition. We also keep around $x^0$ because this encodes not taking anything. Finally, note that $\deg p_r$ is equal to the size of $T$.

Note that there are $n - 1$ total children in $T$, so we do $n - 1$ FFT calculations each of which is bounded by $O(n \log n)$ (size of subtree at most $n$). So, we have the desired $O(n^2 \log n)$ time bound.

2. (25 pts) Min Cost Vertex Cover

You are given a graph $G = (V, E)$ where each vertex $v \in V$ has a nonnegative weight $w_v$. The goal is to pick a subset of the vertices $S$ such that for every edge $e$ in $E$, $e$ has at least one endpoint in $S$. Further, you want to pick $S$ such that the total weight of vertices in $S$ is minimized, that is $\sum_{v \in S} w_v$ is minimized. Throughout this problem we use OPT to refer to the cost of this optimum solution. This problem is NP-hard. Hence we will explore efficient algorithms to approximate the optimum. \footnote{I found it useful to work through this problem with a simple graph of four vertices $\{1, 2, 3, 4\}$, and edges $\{(1, 2), (2, 3), (3, 4)\}$ in mind.}

(a) Express the min weight vertex cover as a linear program. (Hint: you will need a variable $x_v$ for each vertex $v$ and a constraint for every edge, not counting the non-negativity constraints.) Call this $\text{LP}_a$.

Of course $\text{LP}_a$ will not capture the integrality necessary for this problem. But we can solve it by a standard algorithm, and “round” that solution to make it integral. The rounding is done as follows: If $x_v \geq \frac{1}{2}$ it’s rounded to 1. If it’s $< \frac{1}{2}$ it’s rounded to zero. Prove that this rounding scheme results in a feasible solution of cost at most $2\text{OPT}$. \footnote{I found it useful to work through this problem with a simple graph of four vertices $\{1, 2, 3, 4\}$, and edges $\{(1, 2), (2, 3), (3, 4)\}$ in mind.}
(b) Take the dual of LP\(_a\), and call it LP\(_b\). This LP will have a variable for every edge and a constraint for each vertex. (Not counting the non-negativity constraints.) (If your LP\(_b\) seems to have too many variables, take a look at your LP\(_a\). Can you get rid of some constraints that are unnecessary?) Write this LP out.

Now an LP solver can be used to solve LP\(_b\), and find an optimum solution to the dual variables maximizing the objective. Prove that taking the vertices corresponding to the constraints that are tight gives a vertex cover. Also prove that the weight of this cover is at most 2OPT. (Here you will have to make use of weak or strong duality.)

(c) Use the insights from the above parts to devise a third 2-approximation algorithm for this problem that does not have to solve any LP. Its running time should be linear in the number of edges in the graph.

(Hint: your solution can be viewed as constructing a feasible primal solution and a feasible dual solution simultaneously without actually running an LP algorithm. Weak duality then allows you to prove the necessary bound.)

Solution:

(a) Define LP\(_a\) as follows:

$$\text{Minimize } \sum_{v \in V} x_v w_v \quad \text{subject to}$$

$$x_u + x_v \geq 1 \quad \text{for all } \{u, v\} \in E \quad \text{(edge constraints)}$$

$$x_v \geq 0 \quad \text{for all } v \in V.$$

If we restrict each \(x_v\) to be integral, then solving this integer linear program (ILP) leads to a solution to the min-cost vertex cover problem. To see this, consider an optimal solution to the ILP, and let its value be \(\tilde{a}\). Note that we never have \(x_v > 1\) for any \(v\) since otherwise we can lower that variable to \(x_v = 1\), which betters the objective without breaking any constraints. Thus we know \(x_v \in \{0, 1\}\). We can then get a vertex cover by letting \(v \in S\) iff \(x_v = 1\), and this vertex cover has weight equal to \(\tilde{a}\). (This is a vertex cover because for every \(\{u, v\} \in E\), either \(x_u = 1\) or \(x_v = 1\) due to the edge constraints in LP\(_a\).) This gives that OPT \(\leq \tilde{a}\). Similarly, we can convert any vertex cover to a feasible solution to the ILP, and the value of this feasible solution is equal to the weight of the vertex cover. This gives that OPT \(\geq \tilde{a}\). Altogether, we have OPT = \(\tilde{a}\).

Suppose we ignore the integrality constraints and solve LP\(_a\), denoting its optimal value as \(a\). Note that since the integer solution that gave value \(\tilde{a}\) is a feasible solution to LP\(_a\), we have \(a \leq \tilde{a} = \text{OPT}\).

Perform the rounding scheme described in the handout. Let \(\hat{x}_v\) be the rounded value of \(x_v\). Notice that each constraint is still satisfied; if \(x_u + x_v \geq 1\), then either \(x_u \geq 1/2\)

\(^2\)Actually, when taking the dual it’s more natural to think of LP\(_a\) as the “dual” and take its dual to get LP\(_b\), the “primal”, because LP\(_a\) it is a minimization LP.
or \( x_v \geq 1/2 \) and hence one of \( \hat{x}_v \) and \( \hat{x}_u \) is 1. We can once again get a vertex cover by letting \( v \in S \) iff \( \hat{x}_v = 1 \). The weight of this vertex cover is the value of this rounded solution to the LP. Using the fact that \( \hat{x}_v/2 \leq x_v \), we have that the value of this LP solution is

\[
\sum_{v \in V} w_v \hat{x}_v \leq 2 \sum_{v \in V} w_v x_v = 2a \leq 2OPT
\]

as desired.

(b) The dual, LP\(_b\), is as follows:

Maximize \( \sum_{\{u,v\} \in E} y_{\{u,v\}} \) subject to

\[
\sum_{\{u,z\} \in E} y_{\{u,z\}} \leq w_v \quad \text{for all } v \in V
\]

\[
y_{\{u,v\}} \geq 0 \quad \text{for all } \{u,v\} \in E.
\]

Solve this LP and let \( b \) denote its optimal value. Note that in the optimal solution, for every edge \( \{u, v\} \in E \), either the constraint on \( w_u \) or the constraint on \( w_v \) is tight. Otherwise \( y_{\{u,v\}} \) could be increased, which would better the objective. Thus for every edge we add one of the endpoints to our vertex set, giving a vertex cover.

Call our vertex cover \( S \). We know that \( a = b \) due to strong duality, and then we find that the weight of this vertex cover is

\[
\sum_{s \in S} w_s = \sum_{s \in S} \sum_{\{u,s\} \in E} y_{\{u,s\}} \leq \sum_{v \in V} \sum_{\{u,v\} \in E} y_{\{u,v\}} = 2 \sum_{e \in E} y_e = 2b = 2a \leq 2OPT \quad (1)
\]

as desired.

(c) Keep LP\(_b\) in mind. Start with each \( y_{\{u,v\}} = 0 \). Now iterate through each edge \( \{u, v\} \in E \) and increase \( y_{\{u,v\}} \) as much as possible, i.e. until either the constraint on \( w_u \) or the constraint on \( w_v \) is tight. At the end of all of this, as in part (b), we take \( v \) to be in our vertex set \( S \) iff the constraint on \( w_v \) is tight. This once again gives a vertex cover since for each edge \( \{u, v\} \in E \), at least one of \( w_u \) or \( w_v \) is involved in a tight constraint by definition of the algorithm. This is a feasible solution to LP\(_b\), and the weight of this vertex cover satisfies the exact same chain of inequalities as in equation (1). Therefore, this gives a 2-approximation to the min-cost vertex cover problem.

3. (25 pts) Triangle Game

Let \( T = (a, b, c) \) be a triangle in the plane where \( a = (-1, 0) \), \( b = (1, 0) \) and \( c = (0, 2) \). Player I (the “evader”) and Player II (the “searcher”) play the following zero-sum game. The evader picks a point \( X \) in \( T \) and simultaneously the searcher picks a point \( Y \) in \( T \). The payoff to the evader is \( \|X - Y\|^2 \), the square of the Euclidean distance between \( X \) and \( Y \).
Find the value $v$ of this game (from the evader’s viewpoint). Prove that it’s correct by giving a strategy for the evader that achieves at least $v$ in expectation, and one for the searcher that prevents the evader from earning more than $v$ in expectation.

**Solution:** In this game, for both players, a pure strategy is to pick one point, and a mixed strategy is defined by a probability distribution over all the points of the triangle.

We’ll use the approach alluded to right before section 2.3.1 in the notes for lecture 21 on zero sum games. A lower bound $LB$ on the value of the game can be obtained as follows. The maximizing player (the evader in this case) picks some mixed strategy. Given this mixed strategy, the minimizing player picks the *pure* strategy that minimizes the payoff. The result is a lower bound on the payoff for the maximizing player.

Conversely, an upper bound $UB$ can be obtained by specifying some mixed strategy for the minimizing player, then given this, the maximizing player picks a pure strategy that maximizes the payoff.

So the approach to solving this problem is to obtain a lower and an upper bound that are equal, which gives the value of the game.

Let’s start with the upper bound, because that is simpler. An obvious place to start is for the searcher to pick the point in the triangle that is equidistant from the three corners. This is the point $(0,3/4)$ which is $5/4$ away from each of the three corners. After this choice, the evader can do no better than going to any one of the three corners, and obtain a payoff of $25/16$. So we have found an upper bound of $25/16$.

For the lower bound it’s natural to consider distributions that only go to the three corners. Also, by symmetry, let’s limit our search to strategies that give $(-1,0)$ and $(1,0)$ the same probability, $p$, and $(0,2)$ probability $1-2p$. With this strategy, what pure strategy will the searcher pick? And what will the payoff be? Let the point be $(x,y)$. What’s the expected payoff for the evader in this case?

$$f(p, x, y) = p(1 - x)^2 + p(1 + x)^2 + 2py^2 + (1 - 2p)(2 - y)^2.$$

Note that $(1 - x)^2 + (1 + x)^2 = 2 + 2x^2$ is minimized when $x = 0$. What about $y$? What value of $y$ minimizes this?

$$f(p, y) = 2p + 2py^2 + (1 - 2p)(2 - y)^2 = 2p(1 + y^2) + (1 - 2p)(2 - y)^2$$

So given $p$ we want to find the $y$ that minimizes this. (This is what the searcher will do.) Simplifying we get:

$$f(p, y) = -6p + 4 + y^2 - 4y + 8py$$
Now taking the derivative w.r.t. $y$, we get

$$\frac{df}{dy} = 2y - 4 + 8p$$

Setting this to 0 we get $2y = 4 - 8p$, or $y = 2 - 4p$. Let's try a $p$ such that the resulting $y$ value is $3/4$. That is $p = 5/16$.


So with this mixed strategy: with probability $5/16$ go to (-1,0) and with probability $5/16$ go to (1,0) and with probability $6/16$ go to (0,2) the evader obtains an expected value of $25/16$. This is a lower bound on the game. Since it's the same as the upper bound we proved earlier, this is the value of the game.