Homework 5:

Due: October 16, 2025 at 2:30p.m.

This homework must be typed in LaTeX and submitted via Gradescope.

Please ensure that your solutions are complete, concise, and communicated clearly. Use full sentences and plan your presentation before your write. Except where indicated, consider every problem as asking for a proof.

Problem 1. While playing with your pile of n rocks, you begin to wonder about the types of rocks you have. A type of rock is *overly common* if more than half of the rocks you have are of this type.

Unfortunately, you don't have the rock expertise to exactly tell what type a rock is, however, you do have access to a magic machine (i.e. a predicate) that is able to determine if two rocks share the same type.

(a) Describe an algorithm to determine if a collection of n rocks contains an overly common type of rock. If it does, return all of the overly common rocks. Your algorithm should make at most $O(n \log n)$ calls to the magic machine.

(b) Prove the correctness of your algorithm.

(c) Analyze the runtime of your algorithm, justifying that your algorithm makes at most $O(n \log n)$ class to the machine.

Solution. Suppose the pile is split into two halves, P_1 and P_2. If neither of these two subpiles have an overly common rock type, it must be true that there is no overly common type in the entire pile: indeed, for any type t, there are at most $|P_i|/2$ rocks of type t in pile i, so the total number of rocks is $\leq |P_1|/2 + |P_2|/2 = n/2$.

Therefore, we must determine if overly common types exist in each of these halves, suggesting a divide and conquer approach: given two subpiles and whether or not an overly common type is in either pile, we must combine the result to determine if the merged pile has an overly common type.

If neither subpile does, then the answer is no by the argument above. Now suppose at least one of the two subpiles has an overly common rock. Note no subpile can have more than one overly common type. The key idea is that these types are the only candidates for an overly common type in the merged pile. Indeed, suppose the overly common type in the merged pile is not an overly common type in either interval (call this type T). Then, applying similar logic to the first paragraph, at most half of the rocks will be of type T, contradiction.

Thus, we can take a rock of one of the candidate types and compare with every other rock to count how many rocks are of that type. If an overly common type is found, then return all these rocks. Otherwise, there is no overly common type.

To determine time complexity, the recursion comes out to

$$T(n) \leq 2T(n/2) + 2n$$

since a linear search is applied to count the number of rocks resulting in $O(n)$ calls, and there are at most 2 candidates for the overly common type. By Master Theorem, then, the number of calls made is $O(n \log n)$.

Remark: There is a solution that uses $O(n)$ calls: as a start, take pairs of rocks make a call to the machine for each. Depending on the result, what should be done to reduce the search space by at least half? □

Problem 2. Compute the product of the two polynomials using the Fast Fourier Transform (FFT):

- $p(x) = 5x + 2$,
- $q(x) = 6x + 1$.

Specify all (recursive) calls of the FFT algorithm as well as the outputs and the assignments of the temporary variables used during the execution.

Solution. We multiply via convolution of coefficient vectors using FFT of length $N = 4$ (next power of two exceeding $\deg(p) + \deg(q) = 2$).

Coefficient padding

$$p = [\,2,\,5,\,0,\,0\,], \qquad q = [\,1,\,6,\,0,\,0\,].$$

Let $\omega_4 = e^{2\pi i/4} = i$ and $\omega_2 = -1$.

Recursive FFT on p (size 4): Split into evens/odds:

$$p^{(0)} = [2, 0], \quad p^{(1)} = [5, 0].$$

Size-2 FFTs:

$$\mathrm{FFT}_2(p^{(0)}) : \; y_0 = 2 + 0 = 2, \quad y_1 = 2 - 0 = 2 \;\Rightarrow\; [2, 2].$$
$$\mathrm{FFT}_2(p^{(1)}) : \; y_0 = 5 + 0 = 5, \quad y_1 = 5 - 0 = 5 \;\Rightarrow\; [5, 5].$$

Combine to size 4: For $k = 0, 1$,

$$Y_k = E_k + \omega_4^k O_k, \quad Y_{k+2} = E_k - \omega_4^k O_k.$$

With $E = [2, 2]$, $O = [5, 5]$:

$$k = 0 : Y_0 = 2 + 1 \cdot 5 = 7, \quad Y_2 = 2 - 1 \cdot 5 = -3.$$
$$k = 1 : Y_1 = 2 + i \cdot 5 = 2 + 5i, \quad Y_3 = 2 - i \cdot 5 = 2 - 5i.$$

Hence $\mathrm{FFT}_4(p) = [\,7,\; 2 + 5i,\; -3,\; 2 - 5i\,]$.

Recursive FFT on q (size 4): Split:

$$q^{(0)} = [1, 0], \quad q^{(1)} = [6, 0].$$

Size-2 FFTs:

$$\mathrm{FFT}_2(q^{(0)}) : \; [1, 1], \quad \mathrm{FFT}_2(q^{(1)}) : \; [6, 6].$$

Combine to size 4:

$$k = 0 : Z_0 = 1 + 1 \cdot 6 = 7, \quad Z_2 = 1 - 6 = -5.$$
$$k = 1 : Z_1 = 1 + i \cdot 6 = 1 + 6i, \quad Z_3 = 1 - i \cdot 6 = 1 - 6i.$$

Hence $\mathrm{FFT}_4(q) = [\,7,\; 1 + 6i,\; -5,\; 1 - 6i\,]$.

Pointwise product in frequency domain:

$$W_k = Y_k \cdot Z_k :$$

$$\begin{aligned}
W_0 &= 7 \cdot 7 = 49, \\
W_1 &= (2+5i)(1+6i) = 2 + 12i + 5i + 30i^2 = 2 + 17i - 30 = -28 + 17i, \\
W_2 &= (-3) \cdot (-5) = 15, \\
W_3 &= (2-5i)(1-6i) = 2 - 12i - 5i + 30i^2 = 2 - 17i - 30 = -28 - 17i.
\end{aligned}$$

Inverse FFT (size 4): Compute $\mathrm{IFFT}_4(W)$ via the same butterfly with $\omega_4^{-1} = -i$, then divide by 4.
Split W into evens/odds: $W^{(0)} = [49, 15],\ W^{(1)} = [-28 + 17i,\ -28 - 17i]$.
Size-2 IFFTs (which are the same as FFTs up to ω^{-1}, but for size 2 it's identical):

$E = [49{+}15,\ 49{-}15] = [64,\ 34], \quad O = [(-28{+}17i){+}(-28{-}17i),\ (-28{+}17i){-}(-28{-}17i)] = [-56,\ 34i]$.

Combine with $\omega_4^{-1} = -i$:

$$\begin{aligned}
k = 0 : \tilde{Y}_0 &= E_0 + 1 \cdot O_0 = 64 - 56 = 8, \quad \tilde{Y}_2 = E_0 - 1 \cdot O_0 = 64 - (-56) = 120. \\
k = 1 : \tilde{Y}_1 &= E_1 + (-i) \cdot O_1 = 34 + (-i) \cdot (34i) = 34 - 34i^2 = 34 + 34 = 68, \\
\tilde{Y}_3 &= E_1 - (-i) \cdot O_1 = 34 - (-i \cdot 34i) = 34 + 34i^2 = 34 - 34 = 0.
\end{aligned}$$

Finally divide by $N = 4$ to obtain time-domain coefficients:

$$c = \frac{1}{4}\,[\,8,\ 68,\ 120,\ 0\,] = [\,2,\ 17,\ 30,\ 0\,].$$

Thus the product polynomial is

$$p(x)q(x) = 2 + 17x + 30x^2,$$

which matches the direct multiplication $(5x + 2)(6x + 1) = 30x^2 + 17x + 2$.

Temporaries summary (key assignments):

For p :	$p^{(0)} = [2, 0],\ p^{(1)} = [5, 0],\ E = [2, 2],\ O = [5, 5],\ Y = [7,\ 2 + 5i,\ -3,\ 2 - 5i]$.
For q :	$q^{(0)} = [1, 0],\ q^{(1)} = [6, 0],\ E = [1, 1],\ O = [6, 6],\ Z = [7,\ 1 + 6i,\ -5,\ 1 - 6i]$.
Product:	$W = [49,\ -28 + 17i,\ 15,\ -28 - 17i]$.
IFFT prep:	$W^{(0)} = [49, 15],\ W^{(1)} = [-56,\ 34i],\ E = [64, 34],\ O = [-56,\ 34i]$.
Combine:	$\tilde{Y} = [8, 68, 120, 0],\ c = \tilde{Y}/4 = [2, 17, 30, 0]$.

□

Problem 3. Let S and T be two sets of integers in the range $[0, m]$ where m is a power of two. Use a single DFT to compute the following in $O(m \log_2 m)$ time:

- all elements contained in the set

$$S + T := \{s + t : s \in S, t \in T\};$$

- for each element $u \in S + T$, the number $k_u := |\{(s, t) \in S \times T : s + t = u\}|$.

[Hint: Find polynomials p_S and p_T of degree $< m$ that represent the sets S and T. Use x^z to represent item z.]

Solution. Define indicator polynomials of length $N = 2m$ (so that degrees up to $2m - 1$ cover all sums in $[0, 2m]$):

$$p_S(x) = \sum_{s \in S} x^s, \qquad p_T(x) = \sum_{t \in T} x^t.$$

Then the coefficient of x^u in the convolution $p_S \cdot p_T$ equals

$$c_u = \sum_{s+t=u} 1 \;=\; k_u.$$

Hence $u \in S + T$ iff $c_u > 0$, and $k_u = c_u$.

One-forward-DFT We want to avoid two forward DFTs. Pack the two real sequences into one complex sequence $a + ib$:

$$A[j] = \begin{cases} 1 & j \in S \\ 0 & \text{otherwise} \end{cases}, \quad B[j] = \begin{cases} 1 & j \in T \\ 0 & \text{otherwise} \end{cases}, \quad C[j] = A[j] + i\, B[j],$$

for $j = 0, \ldots, N - 1$ with $N = 2m$ (a power of two). Compute a *single* length-N forward DFT:

$$F[k] = \mathrm{DFT}_N(C)[k], \quad k = 0, \ldots, N - 1.$$

By conjugate symmetry of real-input DFTs,

$$\widehat{A}[k] = \frac{F[k] + \overline{F[-k \bmod N]}}{2}, \qquad \widehat{B}[k] = \frac{F[k] - \overline{F[-k \bmod N]}}{2i}.$$

Thus, from F alone (and its conjugate-reversed copy) we can recover both $\widehat{A}$ and $\widehat{B}$ in $O(N)$ time.

For each k,

$$\widehat{C_{\mathrm{conv}}}[k] \;=\; \widehat{A}[k] \cdot \widehat{B}[k].$$

Finally compute one inverse DFT to obtain coefficients $c_0, \ldots, c_{N-1}$:

$$c = \mathrm{IDFT}_N\big(\widehat{C_{\mathrm{conv}}}\big).$$

(Each c_u is real; round to nearest integer to remove numerical error.)

Output:

$$S + T = \{\, u \in [0, 2m] : c_u > 0 \,\}, \qquad k_u = c_u \ \text{ for each } u.$$

Complexity: One forward DFT of length $N = 2m$, $O(N)$ linear work to extract $\widehat{A}, \widehat{B}$ and multiply pointwise, and one inverse DFT of length N:

$$O(N \log N) + O(N) + O(N \log N) = O(m \log m).$$

This uses only *one* forward DFT (instead of two) by packing A and B together, plus the necessary single inverse transform to return to coefficients.

Why $N = 2m$. The largest sum is $\max(S) + \max(T) \leq m + m = 2m$. Choosing $N = 2m$ (a power of two since m is) ensures the circular convolution equals the desired linear convolution without wrap-around. □