Problem 7: Lattice Lights

By Nalin Ranjan

(i) (2 points) The maximum number of shining lights at any distance will occur whenever all the voltages are turned on. So the question is equivalent to asking what the maximum number of lights can be at a fixed distance. We may first observe that once we specify two of its coordinates, the third is determined because we know its taxicab distance from the origin. As we have at most N^2 ways to specify two of the coordinates of a point, the answer cannot be more than N^2.

And indeed, this bound is tight (if we only consider the order of growth). For example, the number of ways we can have the distance be $N-1$ is

$$1+2+3+\cdots+N = N(N+1)/2$$

(The first term counts how many ways we can force the first two coordinates to sum to zero, the second term counts how many ways we can force the first two coordinates to sum to one, etc.) So the answer is $\boxed{\Theta\left(N^2\right)}$.

Scoring Notes: What we were trying to get at in the follow-up questions was that this fact does not imply anything about how hard it is to calculate the statistics we want in problems A and B. Indeed, as we will see, calculating those statistics will not require actually knowing which lattice points have shining lights. From the submissions, though, it seemed as if this wasn't clear, so we awarded full points if you came up with the $\Omega\left(N^2\right)$ bound.

(ii) (3 points) The total number of shining lights is just $|A_x| \cdot |A_y| \cdot |A_z|$, so we just need to calculate the sum of the distances of each shining light. The sum we wish to calculate is

$$\sum_{a\in A_x}\sum_{a\in A_y}\sum_{a\in A_z}[a+b+c] = |A_y|\cdot|A_z|\cdot\text{sum}(A_x) + |A_x|\cdot|A_z|\cdot\text{sum}(A_y) + |A_x|\cdot|A_y|\cdot\text{sum}(A_z)$$

where $\text{sum}(A) = \sum_{a\in A} a$. Putting it all together, we see that the average is simply

$$\frac{\text{sum}(A_x)}{|A_x|} + \frac{\text{sum}(A_y)}{|A_y|} + \frac{\text{sum}(A_z)}{|A_z|}$$

which you can observe is just the sum of the averages of each list. This quantity can be calculated in time linear in $|A_x|$, $|A_y|$, and $|A_z|$, so in the worst case it will take time linear in N. We cannot do better than this without losing generality as the sizes of A_x, A_y, and A_z may be as large as N.

Scoring Notes: For this part, we awarded full credit for a linear solution, 2 points for a quadratic solution, and 1 point for anything else that was correct.

(iii) (10 points) The best algorithm we know runs in $O(N \log N)$ time, and it is likely that this is optimal. Note that for a given distance D, the number of lights at that distance that are shining is

$$\sum_{a=0}^{\min(D,N-1)} \mathbb{1}_{a \in A_x} \sum_{b=0}^{D-a} \mathbb{1}_{b \in A_y} \cdot \mathbb{1}_{(D-a-b) \in A_z}$$

If we look closely, this sum is the Dth element of the convolution of the three vectors v_x, v_y, and v_z, where

$$v_x[k] = \mathbb{1}_{k \in A_x}, \; v_y[k] = \mathbb{1}_{k \in A_y}, \; v_z[k] = \mathbb{1}_{k \in A_z}$$

The convolution can be calculated efficiently using the Fast Fourier Transform, as the convolution theorem states that

$$\mathcal{F}(x) \cdot \mathcal{F}(y) = \mathcal{F}(x * y)$$

where $\mathcal{F}$ represents the Fourier Transform, $*$ represents the convolution, and $\cdot$ represents point-wise multiplication between the vectors. Put all together, an algorithm that solves this problem in $O(N \log N)$ operations could be as follows:

1. Calculate the bit vectors v_x, v_y, and v_z as defined above. Since D may be as large as $3N - 3$, pad these vectors with zeros until they have length $3N - 2$.
2. Calculate $\mathcal{F}(v_x)$, $\mathcal{F}(v_y)$, and $\mathcal{F}(v_z)$ by applying the Fast Fourier Transform on the respective vectors.
3. Calculate $\mathcal{F}(v_x * v_y * v_z)$, which by the Convolution Theorem is just the point-wise product of the three vectors calculated in the previous step.
4. Apply the inverse Fast Fourier Transform on the result from step 3.
5. However you like, find the indices of the C largest entries. This can be done in many ways in $O(N \log N)$ time, e.g. by sorting.

Any solution that has the same time complexity likely will also have to utilize the Fast Fourier Transform. As part (i) tells us, any solution that can be trivially adapted to let us know exactly which lights at a given distance are shining is no good, as this number could be as great as $\Theta(n^2)$.

Scoring Notes: We awarded full credit for a linearithmic solution, 5 points for a quadratic solution, 3 points for a cubic solution, and 2 points for anything else that was correct.

What's the moral of the story for all of these questions? It's that even if you have to count a potentially large number of objects, there may be efficient ways to do so if you are given a compressed representation of them. (Here, we didn't enumerate all of the lights which were shining; instead, we gave you three arrays of length $O(N)$ gave you all the information you needed.) Always take a moment to see if a lazy approach can solve your problem, even when (as in this case) it's not even clear if such an approach should exist!

Plaudits:

- Congratulations to Kiril Bangachev '22 and Rahul Saha '22, Seyoon Ragavan '24 and Lucas Salvador '20 (now grad), Antonio Molina (grad), and Calvin Deng (grad) for finding the elusive FFT solution on the last part!
- Further praise goes to Antonio and Seyoon/Lucas for providing very detailed solutions to the problem. They addressed all of the small details, including the issue of arithmetic precision - something that goes above and beyond what was required to get full credit. And on top of that, they submitted beautifully formatted Latex files.