

6.046
PROBLEM SET 3

JOHN WANG

Collaborators: Ryan Liu

1. PROBLEM A

Problem 1.1. Describe a simple $O(m^2n)$ time algorithm to solve the electric potential problem.

Solution The simple algorithm will iterate over all grid points not occupied by a charge, and calculate the sum $\sum_{i=1}^n \frac{q_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}}$ by iterating over all n grid points with a charge. We will write pseudo-code and then analyze its runtime. First, assume that we have the following input data in matrix form:

$$(1.1) \quad \begin{pmatrix} q_{11} & \cdots & q_{1m} \\ \vdots & \ddots & \vdots \\ q_{m1} & \cdots & q_{mm} \end{pmatrix}$$

Here, $q_{ij} = 0$ if point (i, j) in the grid is not occupied by a charge. Otherwise, q_{ij} denotes the value of the charge at point (i, j) on the gride. Using this input data, the algorithm will have the following pseudocode:

```
def find_electric_potential(matrix, m):
    occupied = []
    unoccupied = []
    for i in xrange(m):
        for j in xrange(m):
            if matrix[i][j] != 0:
                occupied.append((i, j, matrix[i][j]))
            else:
                unoccupied.append((i, j))

    output = []
    for (x, y) in unoccupied:
        sum = 0
        for (xi, yi, qi) in occupied:
            sum += qi / ((x - xi)^2 + (y - yi)^2)^(0.5)
        output.append((x, y, sum))
    return output
```

The algorithm returns the list of tuples: $[(x_1, y_1, V(x_1, y_1)), \dots, (x_{m-n}, y_{m-n}, V(x_{m-n}, y_{m-n}))]$, where $V(x_i, y_i)$ denotes the potential at (x_i, y_i) . The algorithm has a runtime of $O(m^2n)$ because the first block of for loops must iterate over m^2 items. The second block of for loops over $m^2 - n$ items, and performs n operations for each of these items. Thus, the total runtime is $O(m^2) + O((m^2 - n)n)$. Simplifying this, we obtain: $O(m^2 + m^2n - n^2) = O(m^2n)$ since $m^2 \geq n^2$.

The correctness of this algorithm follows immediately from the definition of the problem. We compute $V(x_i, y_i)$ exactly as $\sum_{i=1}^n \frac{q_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}}$ for each of the unoccupied points. \square

2. PROBLEM B

Problem 2.1. Let $Z_{2m-1} = 0, 1, \dots, 2m-2$. Find two functions $f, g : Z_{2m-1} \times Z_{2m-1} \rightarrow \mathbb{R}$ such that the potential at (x, y) for $0 \leq x, y < m$ equals the convolution of f and g :

$$\begin{aligned} V(x, y) &= (f \otimes g)(x, y) \\ &= \sum_{x'=0}^{2m-2} \sum_{y'=0}^{2m-2} f(x', y') \cdot g(x - x', y - y') . \end{aligned}$$

Importantly, in this definition $x - x'$ and $y - y'$ are computed in the additive group \mathbb{Z}_{2m-1} , which is a fancy way of saying they are computed modulo $2m - 1$.

Solution We shall take the following functions:

$$(2.1) \quad f(x, y) = \begin{cases} 0 & \text{if } x > m - 1 \text{ or } y > m - 1 \\ \frac{q(x, y)}{4\pi\epsilon_0} & \text{otherwise} \end{cases}$$

And also we take:

$$(2.2) \quad g(x, y) = \begin{cases} 0 & \text{if } x = 0, y = 0 \\ (\bar{x}^2 + \bar{y}^2)^{-1/2} & \text{otherwise} \end{cases}$$

Where \bar{x} and \bar{y} are defined as follows:

$$(2.3) \quad \bar{x} = \begin{cases} x & \text{if } x < m \\ 2m - 1 - x & \text{if } x \geq m \end{cases} \quad \bar{y} = \begin{cases} y & \text{if } y < m \\ 2m - 1 - y & \text{if } y \geq m \end{cases}$$

We know that $x - x'$ ranges from $-(m - 1)$ to $(m - 1)$. However, we know that $-(m - 1) \equiv -(m - 1) + 2m - 1 \equiv m \pmod{2m - 1}$. Thus, if $x - x' > m$, all we need to do is subtract x from $2m - 1$ to obtain a correct mapping. The same can be done for $y - y'$.

We shall show that these two functions satisfy $V(x, y) = (f \otimes g)(x, y)$. This can be shown by simply expanding the definitions. We know the following:

$$(2.4) \quad (f \otimes g)(x, y) = \sum_{x'=0}^{2m-2} \sum_{y'=0}^{2m-2} f(x', y') g(x - x', y - y')$$

$$(2.5) \quad = \sum_{x'=0}^{m-1} \sum_{y'=0}^{m-1} \frac{q(x', y')}{4\pi\epsilon_0} g(x - x', y - y')$$

$$(2.6) \quad = \sum_{x'=0}^{m-1} \sum_{y'=0}^{m-1} \frac{q(x', y')}{4\pi\epsilon_0} \frac{1}{\sqrt{(x - x')^2 + (y - y')^2}}$$

$$(2.7) \quad = \sum_{i=1}^n \frac{q_i}{4\pi\epsilon_0 \sqrt{(x - x_i)^2 + (y - y_i)^2}}$$

$$(2.8) \quad = V(x, y)$$

Where the last part follows since $q(x', y') = 0$ unless it holds a charge q_i . Thus, we have chosen functions f and g which satisfy the property that their convolutions lead to the electric potential. \square

3. PROBLEM C

Problem 3.1. Prove that for any two functions $f, g : \mathbb{Z}_k \times \mathbb{Z}_k \rightarrow \mathbb{R}$ and for any point $(a, b) \in \mathbb{Z}_k \times \mathbb{Z}_k$, we have

$$\widehat{(f \otimes g)}(a, b) = k^2 \cdot \widehat{f}(a, b) \cdot \widehat{g}(a, b).$$

Solution First, we shall expand $k^2 \widehat{f}(a, b) \cdot \widehat{g}(a, b)$ using the definition of the discrete fourier transform:

$$(3.1) \quad k^2 \cdot \widehat{f}(a, b) \cdot \widehat{g}(a, b) = \frac{1}{k^2} \left(\sum_{x=0}^{k-1} \sum_{y=0}^{k-1} f(x, y) w_k^{-ax-by} \right) \left(\sum_{x'=0}^{k-1} \sum_{y'=0}^{k-1} g(x', y') w_k^{-ax'-by'} \right)$$

$$(3.2) \quad = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \sum_{x'=0}^{k-1} \sum_{y'=0}^{k-1} f(x, y) g(x', y') w_k^{-a(x+x')-b(y+y')}$$

Now, we want to see what $\widehat{(f \otimes g)}(a, b)$ is equal to. We shall expand this expression according to the definitions of convolution and discrete fourier transforms:

$$(3.3) \quad \widehat{(f \otimes g)}(a, b) = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \widehat{(f \otimes g)}(x, y) w_k^{-ax-by}$$

$$(3.4) \quad = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \left(\sum_{x'=0}^{k-1} \sum_{y'=0}^{k-1} f(x', y') g(x - x', y - y') \right) w_k^{-ax-by}$$

Notice that the sum inside the parentheses can be broken up into two sums. The first group shall consist of x, x', y, y' combinations such that $x - x' > 0$, $x - x' < k - 1$, $y - y' > 0$, and $y - y' < k - 1$. The second group shall consist of x, x', y, y' combinations such that $x - x' < 0$, $x - x' > -(k - 1)$, $y - y' < 0$, and $y - y' > -(k - 1)$. However, since $x - x'$ and $y - y'$ are taken modulo $k - 1$, we see that these two groups yield the same sums. Writing this out, we have:

$$(3.5) \quad = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \left(\sum_{x'=x-(k-1)}^x \sum_{y'=y-(k-1)}^y f(x', y') g(x - x', y - y') \right.$$

$$(3.6) \quad + \left. \sum_{x'=x}^{x+(k-1)} \sum_{y'=y}^{y+(k-1)} f(x', y') g(x - x', y - y') \right) w_k^{-ax-by}$$

Since the set $\{0, 1, \dots, k-1\}$ is congruent to $\{-(k-1), -(k-1)+1, \dots, -1\}$ modulo $k-1$, we see that the sums in between the parentheses in the above equation can be combined into:

$$(3.7) \quad = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \sum_{\bar{x}=0}^{k-1} \sum_{\bar{y}=0}^{k-1} f(x', y') g(\bar{x}, \bar{y}) w_k^{-a(\bar{x}+x')-y(\bar{y}+y')}$$

Where we have made the transformation $\bar{x} = x - x'$ and $\bar{y} = y - y'$. Looking back at our expanded expression of $k^2 \hat{f}(a, b) \cdot \hat{g}(a, b)$, it is clear that we have shown $(\widehat{f \otimes g})(a, b) = k^2 \cdot \hat{f}(a, b) \cdot \hat{g}(a, b)$ to be true. \square

4. PROBLEM D

Problem 4.1. Design an $O(k^2 \lg k)$ time algorithm to compute the discrete Fourier transform and its inverse.

Solution For a function $h(x, y) : \mathbb{Z}_k \times \mathbb{Z}_k \rightarrow \mathbb{C}$, we would like to compute the discrete Fourier transform defined as the following:

$$(4.1) \quad \hat{h}(a, b) = \frac{1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} h(x, y) w_k^{-ax-by}$$

$$(4.2) \quad = \frac{1}{k^2} \sum_{x=0}^{k/2-1} \sum_{y=0}^{k/2-1} \left(h(2x, 2y) w_k^{-2ax-2by} + h(2x+1, 2y+1) w_k^{-a(2x+1)-b(2y+1)} \right. \\ \left. + h(2x, 2y+1) w_k^{-2ax-b(2y+1)} + h(2x+1, 2y) w_k^{-a(2x+1)-2y} \right)$$

$$(4.3) \quad = \frac{1}{k^2} \sum_{x=0}^{k/2-1} \sum_{y=0}^{k/2-1} h(2x, 2y) w_{k/2}^{-ax-by} + \frac{w_k^{-a-b}}{k^2} \sum_{x=0}^{k/2-1} \sum_{y=0}^{k/2-1} h(2x+1, 2y+1) w_{k/2}^{-ax-by} \\ + \frac{w_k^{-b}}{k^2} \sum_{x=0}^{k/2-1} \sum_{y=0}^{k/2-1} h(2x+1, 2y) w_{k/2}^{-ax-by} + \frac{w_k^{-a}}{k^2} \sum_{x=0}^{k/2-1} \sum_{y=0}^{k/2-1} h(2x, 2y+1) w_{k/2}^{-ax-by}$$

The last equation comes from the fact that $w_k^2 = w_{k/2}$. We have split the problem up into finding the discrete Fourier transform for $h(a, b)$ where a, b are even, a, b are odd, and a, b are alternating even and odd. However, we have decreased our space by half, so that we are left with much less work.

The only difference is that there is a sign change in the last expression to accomodate the fact that a and b are in $\{k/2, \dots, k-1\}$. Using this as the mathematical background of our algorithm, we can now develop an $O(k^2 \lg k)$ time algorithm for computing the discrete fourier transform. Since we want to compute $\hat{h}(a, b)$, and we have shown that $\hat{h}(a, b)$ can be broken apart into two smaller problems, we can use a recursive, divide and conquer approach. The pseudo-code is given below:

```
def dft(q, k):
    if k = 1:
        return q[0][0]
    else:
        Initiate Q as a k by k output matrix
        out1 = dft(q[x][y] where x,y are even, k/2)
        out2 = dft(q[x][y] where x,y are odd, k/2)
        out3 = dft(q[x][y] where x is even and y is odd, k/2)
        out4 = dft(q[x][y] where x is odd and y is even, k/2)
```

```

for x in xrange(k/2-1):
    for y in xrange(k/2-1):
        Q[x][y] = (out1[x][y] + out2[x][y]*unityroots(k,-a-b)
                  + out3[x][y]*unityroots(k,-b)
                  + out4[x][y]*unityroots(k,-a))/k^2
    Q[x+k/2][y+k/2] = Q[x][y]
return Q

```

The inverse can be computed similarly, but by dropping the $\frac{1}{k^2}$ term. The code for the inverse is so close to the code for the dft function that we will not write it again. Moreover, it is clear that by changing the multiplicative term, we do not change the runtime.

For an analysis of the runtime, we start with a problem of size k . In the algorithm, we run $dft(q, k/2)$ four times, then we perform $O(k^2)$ arithmetic operations (we will say that addition, multiplication, and division all take $O(1)$). Then, we have the recurrence:

$$(4.4) \quad T(k) = 4T(k/2) + ck^2$$

$$(4.5) \quad = ck^2 + 4(c(k/2)^2 + 4T(k/4))$$

$$(4.6) \quad = ck^2 + ck^2 + 16(c(k/4)^2 + T(k/8)^2)$$

$$(4.7) \quad = \underbrace{ck^2 + \dots + ck^2}_{\lg k}$$

$$(4.8) \quad = ck^2 \lg k$$

Therefore, the running time is $T(k) = O(k^2 \lg k)$, which is what we wanted. We have already shown correctness since we are using equation (4.2) to compute $\hat{h}(a, b)$ in our algorithm. We have only rearranged terms in $\hat{h}(a, b)$ and used the fact that $w_k^2 = w_{k/2}$ to design our algorithm. The proof of correctness follows from equation (4.2), assuming that k is a power of 2.

The diagram below shows the intricacies of the algorithm. Computing the discrete Fourier transform involves breaking it apart into four smaller pieces. These consist of indices i, j such that they are even/even, even/odd, odd/even, and odd/odd. The diagram below shows how one would split up a 4×4 matrix into different subproblems.

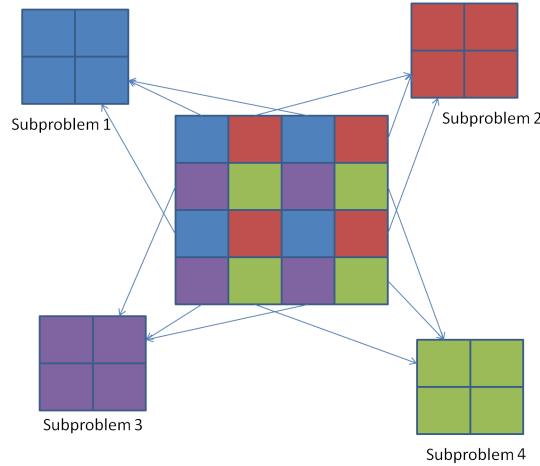


FIGURE 1. Diagram of Subproblem Partitioning

□

5. PART E

Problem 5.1. Design an $O(m^2 \lg m)$ time algorithm to solve the electric potential problem for a grid of size $m \times m$.

Solution

We will use the $O(k^2 \lg k)$ algorithm to compute the discrete Fourier transform and the two functions f and g defined in part b to complete this algorithm. First, we know that $V(x, y) = (f \otimes g)(x, y)$. Moreover,

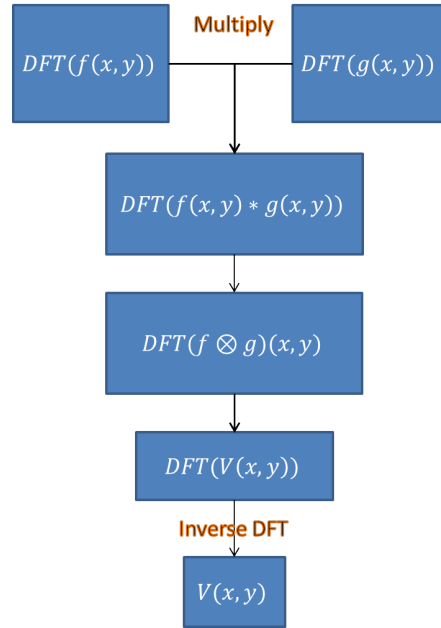


FIGURE 2. Flow Chart of the Algorithm

we know that $(\widehat{f \otimes g})(x, y) = \widehat{f}(x, y)\widehat{g}(x, y)$ by part *c*. Thus, we know that $\widehat{V}(x, y) = \widehat{f}(x, y)\widehat{g}(x, y)$. Thus, in order to find $V(x, y)$, we can simply take the inverse discrete Fourier transform of $\widehat{f}\widehat{g}$. Denote the inverse discrete Fourier transform of a function f by $dft^{-1}(f)$. We know the following:

$$(5.1) \quad V(x, y) = dft^{-1}(\widehat{f}(x, y)\widehat{g}(x, y))$$

However, we already have an algorithm that computes $\widehat{f}(x, y)$ and $\widehat{g}(x, y)$ in $O(k^2 \lg k)$, where k is size of the $k \times k$ space. However, since $f, g : \mathbb{Z}_m \times \mathbb{Z}_m \rightarrow \mathbb{C}$, we see that $k = m$ and the algorithm runs in $O(m^2 \lg m)$. Moreover, we can compute $dft^{-1}(\widehat{f}(x, y)\widehat{g}(x, y))$ in $O(m^2 \lg m)$ because the inverse discrete Fourier transform takes the same time as the discrete Fourier transform, and multiplication takes $O(1)$ time. Therefore, we see that we can compute $V(x, y)$ in a total time of $O(m^2 \lg m)$.

The algorithm, then, is as follows: take the discrete Fourier transform of $f(x, y)$ and $g(x, y)$ as defined in part *b* on the space $\mathbb{Z}_m \times \mathbb{Z}_k$. Then, multiply the together, and take the inverse discrete Fourier transform of the product. This should give the electric potential at all points (x, y) where $x, y \in \{0, \dots, m-1\}$. To finish the problem, we return the potentials at the points (x_i, y_i) which are not occupied by a charge.

The correctness of this algorithm follows from the argument given in the first paragraph. It relies on the observations pointed out in parts *b*, *c*, and *d*. A diagram/example of the algorithm is given in the figure 2.

□