COL351
Assignment 3

Mallika Prabhakar (2019CS50440)
Sayam Sethi (2019CS10399)

September 2021

Contents

1 Question 1

Question 1

Question. *Question*

Solution.

Claim 1.1.

Proof. □

Algorithm 1 DP solution for partitioning

1:

□

2 Question 2

Question 2

Question. *The total net force on particle j, by Coulomb's Law, is equal to*

$$F_j = \sum_{i<j} \frac{Cq_iq_j}{(j-i)^2} - \sum_{i>j} \frac{Cq_iq_j}{(j-i)^2} \tag{1}$$

Design an algorithm that computes all the forces F_j in $O(n \log n)$ time.

Solution. We will use polynomial multiplication to solve this question. Consider the polynomials:

$$\begin{aligned} A(x) &= (0, q_1, q_2, \ldots, q_n) \\ B(x) &= \left(-\frac{1}{(n-1)^2}, -\frac{1}{(n-2)^2}, \ldots, -\frac{1}{1^2}, 0, \frac{1}{1^2}, \ldots, \frac{1}{(n-2)^2}, \frac{1}{(n-1)^2}\right) \end{aligned} \tag{2}$$

In the above representation, only the coefficients of $A(x), B(x)$ are shown. The degrees of $A(x)$ and $B(x)$ are n and $2n-2$ respectively. Now, in the product $P(x) = A(x) \cdot B(x)$, consider the coefficient of x^{j+n-1}. To visualise this, we will write the polynomials as:

$$\begin{array}{lllllll} A(x) = & & q_n x^n + \cdots + & q_{j+1}x^{j+1} + & q_j x^j + & q_{j-1}x^{j-1} + \cdots + q_1 x^1 + 0x^0 \\ B(x) = \cdots + & -\frac{1}{(n-j)^2}x^{j-1} + \cdots + & -\frac{1}{1^2}x^{n-2} + & 0x^{n-1} + & \frac{1}{1^2}x^n + \cdots + \frac{1}{(j-1)^2}x^{j+n-2} \\ & & & & & + \cdots \end{array} \tag{3}$$

Multiplication of corresponding terms gives terms with power of x as $j+n-1$, and thus formally, the coefficient of x^{j+n-1} can be written as:

$$P(x)[j+n-1] = \sum_{k=1}^{n-j} q_{j+k} \cdot -\frac{1}{k^2} + 0 + \sum_{k=1}^{j-1} q_{j-k} \cdot \frac{1}{k^2} \tag{4}$$

Where $P(x)[p]$ denotes the coefficient of x^p in $P(x)$.
Equation 4 can be rewritten as:

$$\begin{aligned} P(x)[j+n-1] &= \sum_{i=j+k,k=1}^{n-j} q_i \cdot -\frac{1}{(j-1)^2} + \sum_{i=j-k,k=1}^{j-1} q_i \cdot \frac{1}{(j-1)^2} \\ &= -\sum_{i=j+1}^{n} \frac{q_i}{(j-1)^2} + \sum_{i=1}^{j-1} \frac{q_i}{(j-1)^2} \\ &= \sum_{i<j} \frac{q_i}{(j-1)^2} - \sum_{i>j} \frac{q_i}{(j-1)^2} \\ &= \frac{F_j}{Cq_j} \\ \implies F_j &= P(x)[j+n-1] \times Cq_j \end{aligned} \tag{5}$$

Therefore, we have derived an alternate method for computing F_j. Since this involves computing product of polynomials, we can perform the polynomial product in $O(n \log n)$ since both $A(x), B(x)$ are polynomials of degree $O(n)$. Once $C(x)$ has been computed, we can then compute F_j in $O(1)$ for each j by dividing the corresponding coefficient with Cq_j. The exact algorithm is given as:

Algorithm 2 Computing F_j for $j \in \{1, 2, \ldots, n\}$

1: **procedure** COMPUTE FORCES((q, n))
2: $\quad A \leftarrow q$
3: $\quad B \leftarrow \left[-\frac{1}{(n-1)^2}, -\frac{1}{(n-2)^2}, \ldots, -\frac{1}{1^2}, 0, \frac{1}{1^2}, \ldots, \frac{1}{(n-2)^2}, \frac{1}{(n-1)^2}\right]$
4: $\quad P \leftarrow \mathit{multiply}(A, B)$ ▷ multiply $A(x)$ and $B(x)$ using FFT "divide and conquer" algo
5: $\quad F \leftarrow C[n : 2n-1]$ ▷ taking subarray corresponding to coefficients of x^{j+n-1}
6: $\quad$ **for** $i \in [1, n]$ **do** ▷ 1-indexed array
7: $\quad\quad F[i] \leftarrow F[i] \times Cq[i]$
8: $\quad$ **end for**
9: $\quad$ **return** F
10: **end procedure**

Proof of Correctness: The proof of correctness of the *FFT Algorithm* has been discussed in the lectures. The correctness of lines $5-8$ has been proved from Equation 5.

Time Complexity: All operations except the *FFT Algorithm* are $O(n)$ operations. The complexity of *FFT Algorithm* has been shown to be $O(d \log d)$ where d is the degree of the polynomial. Since the degrees of $A(x), B(x)$ are $O(n)$, the *FFT Algorithm* can be computed in $O(n \log n)$ time.

Therefore, all forces F_j can be computed in $O(n \log n)$ time. This completes the design of the algorithm along with proof of correctness and time complexity. $\square$

3 Question 3

3.1 3.1

Question 3

Question. *Prove that the graph $H = (V, E_H)$ can be computed from G in $O(n^\omega)$ time, where ω is the exponent of matrix-multiplication.*

Solution. $\square$

3.2 3.2

Question 1

Question. *Question*

Solution.

Claim 3.1.

Proof. $\square$

Algorithm 3 DP solution for partitioning

1:

$\square$

3.3 3.3

Question 1

Question. *Question*

Solution.

Claim 3.2.

Proof. $\square$

Algorithm 4 DP solution for partitioning

1:

$\square$

3.4 3.4

Question 1

Question. *Question*

Solution.

Claim 3.3.

Proof. □

Algorithm 5 DP solution for partitioning

1:

□

3.5 3.5

Question 1

Question. *Question*

Solution.

Claim 3.4.

Proof. □

Algorithm 6 DP solution for partitioning

1:

□

4 Question 4

4.1 4.1

Question 1

Question. *Question*

Solution.

Claim 4.1.

Proof. □

Algorithm 7 DP solution for partitioning

1:

□

4.2 4.2

Question 1

Question. *Question*

Solution.

Claim 4.2.

Proof. □

Algorithm 8 DP solution for partitioning

1:

□

4.3 4.3

Question 1

Question. *Question*

Solution.

Claim 4.3.

Proof. □

Algorithm 9 DP solution for partitioning

1:

$\square$