7

CS345A: Design and Analysis of Algorithms

Quiz 1

Marks = 11

Date: 19 August 2023

NAME: DIVYANSH

ROLL No: 210355

Attempt any one of the following problems.

1. Easy Problem (*7 marks*)

(a) *(marks=3)*

Consider two sets A and B, each having n integers in the range from 0 to $10n$. We wish to compute the **Cartesian** sum of A and B, defined by

$$C = \{x + y \ : \ x \in A \text{ and } y \in B\}.$$

Note that the integers in C are in the range from 0 to $20n$. We want to find the elements of C and the number of times each element of C is realized as a sum of elements in A and B. You need to design an $O(n \log n)$ time algorithm for this problem by converting it into an instance of polynomial multiplication problem. In the following box, describe two polynomials $P_1(x)$ and $P_2(x)$, in terms of A and B respectively, that need to be multiplied. Also state how you will get the solution of the original problem from the product $P_1(x) \times P_2(x)$.

Answer: Let the set $A := \langle a_1, a_2, .., a_n \rangle$ and $B := \langle b_1, b_2, .., b_n \rangle$

Now, I will define $P_1(x) := x^{a_1} + x^{a_2} + x^{a_3} + \dots + x^{a_n}$ ✓

and, $P_2(x) := x^{b_1} + x^{b_2} + x^{b_3} + \dots + x^{b_n}$ ✓

which needs to be multiplied. $P_1(x) \times P_2(x)$ multiplication can be obtanied in $O(n \log n)$. The elements of C will be the power of each x terms and its coefficients will give the number of times it is realized. ✓ 3

(b) *(2 marks)*

Refer to the Gayle Shapley algorithm discussed in the class. Recall that it was 'man proposing' version. Provide arguments in the following box to justify that the number of iterations of the while-loop in the algorithm is $O(n^2)$.

Answer: Let S be the set of man, $|S| = n$ and $L(m_i)$ be the preference list of each man, $|L(m_i)| = n$ for each i.

For each iteration in the while loop, either $|S|$ decreases ✓ or a woman is removed from $L(m_i)$ $\{|L(m_i)|$ decreases$\}$ ✓ 2

hence this lead to $O(n^2)$ $\{n$ times decrease in $|S|$ and in $|L(m_i)|.\}$

(c) *(2 marks)*
You need to provide an instance of the stable matching problem on a set of 4 men and 4 women such that the output of the 'man proposing' version of the Gayle Shapley algorithm is exactly the same as the output of the 'woman proposing' version.

Let W_1, W_2, W_3, and W_4 be the four women and M_1, M_2, M_3, and M_4 be the four men.

Fill in the preference lists of women for the desired instance.

	1	2	3	4
W_1	M_1	M_2	M_3	M_4
W_2	M_2	M_1	M_3	M_4
W_3	M_3	M_1	M_2	M_4
W_4	M_4	M_1	M_2	M_3

Fill in the preference lists of men for the desired instance.

	1	2	3	4
M_1	W_1	W_2	W_3	W_4
M_2	W_2	W_1	W_3	W_4
M_3	W_3	W_1	W_2	W_4
M_4	W_4	W_1	W_2	W_3

State below the stable matching produced by the Gayle Shapley algorithm for the instance given above. Remember that it has to be the same for both the versions.

output of 'man proposing' version :=

$\{\langle M_1, W_1\rangle, \langle M_2, W_2\rangle, \langle M_3, W_3\rangle, \langle M_4, W_4\rangle\}$

output of 'woman proposing' version :=

$\{\langle W_1, M_1\rangle, \langle W_2, M_2\rangle, \langle W_3, M_3\rangle, \langle W_4, M_4\rangle\}$

✓ 2

2. Hard Problem *(11 marks)*

A rectangle is orthogonal if each of its sides is parallel to the x-axis or y-axis. For any 4 numbers a, b, c, and d with $a < b$ and $c < d$, let *Rectangle*(a, b, c, d) denote the orthogonal rectangle defined by the intersection of 4 regions: $x \geq a$, $x \leq b$, $y \geq c$, and $y \leq d$.

Let P be a set of n points in 2-D plane. Assume, without loss of generality, that no two points have the same x-coordinates or y-coordinates. There is an $O(n)$ size static data structure for P that can efficiently report all points from P that are present inside any query *Rectangle*(x_1, x_2, y_1, y_2).

This data structure is just a binary tree $T(P)$ consisting of n nodes. Each node $\nu \in T(P)$ stores a unique point from P, denoted by $p(\nu)$. Let $P(\nu)$ denote the set of points from P that belong to the subtree rooted at ν. ν also stores an orthogonal rectangle denoted by $R(\nu)$ such that $P(\nu)$ lies inside $R(\nu)$. Note that $p(\nu)$ and $R(\nu)$ are the only fields stored at ν in addition to the pointers to its left child, right child, and parent.

For any node ν in $T(P)$, let ν_ℓ and ν_r denote respectively the left child and the right child of ν.

(a) *(marks=2, 2, 1) Description of the data structure*
Let ν be any arbitrary node in $T(P)$, and let i be its depth (distance from the root) in the tree. Describe $P(\nu_\ell)$ in the following box.

Answer: Let for node v, $R(v) := (x_1, x_2, y_1, y_2)$
So, $P(V_\ell)$ denotes the points in the region denoted by $x \geq x_1$ and $x < p(v)$. α 0

Describe $p(\nu_r)$ in the following box.

Answer: Consider set $P :=$ points in $P(V)$ whose x-coordinates are greater that $p(v)$.
$p(v_r) :=$ x-median of P. α 0

Describe $R(\nu_r)$ in the following box.

Answer: Let $R(v) := (x_1, x_2, y_1, y_2)$
So $R(v_r) := (p(v), x_2, y_1, y_2)$ α 0

Note: The answers to the above questions have to be in terms of the parameters associated with node ν.

(b) *(marks=3)*
Let **Report_Points**$(x_1, x_2, y_1, y_2, \nu)$ be an efficient recursive procedure to report all those points from $P(\nu)$ that lie inside *Rectangle*(x_1, x_2, y_1, y_2). Write a neat pseudocode for this procedure in the following box.

Answer:

```
if (v is leaf node) {   // X(p(v)) denotes x-coordinate of p(v).
        if (p(v) is in Rectangle(x1, x2, y1, y2)) print (p(v));
  }
else { if (x2 < X(p(v0))) Report_Points (x1, x2, y1, y2, vl);
   else if (x1 > X(p(v0))) Report_Points (x1, x2, y1, y2, vR);
   else { Report_Points (x1, X(p(v)), y1, y2, vl);
          Report_Points (X(p(v)), x2, y1, y2, vR);
}}
```

0

Insufficient

(c) *(marks=3)*
Let ν_0 denote the root node of $T(P)$. State asymptotically tight bound on time complexity of **Report_Points**$(x_1, +\infty, -\infty, +\infty, \nu_0)$, for any given number x_1. Your answer has to be in terms of n and k only, where k is the number of points belonging to the query rectangle. You must also provide a brief justification for the same (no marks for an answer without justification).

Answer: $O(\log(n))$ if $K > n/2$ and $O(\log(K))$ if $K < n/2$ 0

Justification: Argument: for any set of n Points, the Report_Points function goes recursively from root to leaf having $\log(n)$ time (height). Now is $K > n/2$ which means that x_1 is in left half so, $T(n/2) + T(K - n/2)$ this will lead to $O(\log n)$ for $K < n/2$, (x_1 is in right half with K points) so $O(\log(K))$.

Note:
The student whose aim is beyond getting A^* in this course should, some day after 20th August, try to analyse the time complexity of **Report_Points**$(x_1, x_2, y_1, y_2, \nu_0)$ for any $x_1 < x_2$ and $y_1 < y_2$, and try to compare it with the time complexity of **Report_Points**$(x_1, +\infty, -\infty, +\infty, \nu_0)$.

Hint for the Hard Problem: For the root node ν_0 in tree $T(P)$, $p(\nu_0)$ is the x-median of P, and $R(\nu_0)$ is *Rectangle*$(-\infty, +\infty, -\infty, +\infty)$.