Homework 3

Date: 2023-07-04 15:09
type: #Homework
Class: Com s 311

Divide & Conquer Paradigm Practice

Question 1

Solve the following recurrences. You must show the steps of solving recurrence. You may not use the Master Theorem.

1. $T(n) = T\left(\frac{n}{4}\right) + cn, T(1) = 1$

$$T(n) = T\left(\frac{n}{4^2}\right) + \frac{cn}{4} + cn = \underbrace{T\left(\frac{n}{4^3}\right) + \frac{cn}{4^2} + \frac{cn}{4} + cn}_{\text{Continued pattern}} \implies T(n) = cn + \frac{cn}{4} + \frac{cn}{4^2} + \cdots + \frac{cn}{4^k}$$

$$= cn\left(1 + \frac{1}{4} + \frac{1}{4^2} + \cdots + \frac{1}{4^k}\right) \to T(n) = cn\underbrace{\left(\frac{1\left(1 - \frac{1}{4}^k\right)}{1 - \frac{1}{4}}\right)}_{\text{infinite sum equality}}$$

$$\frac{n}{4^k} = 1 \to \log n = k \log 4 \to k = \log_4 n$$

$$T(n) = cn\left(\frac{1 - \frac{1}{4^{\log_4 n}}}{1 - \frac{1}{4}}\right) = cn\left(\frac{1 - \frac{1}{n}}{\frac{3}{4}}\right) = \frac{4}{3}cn\left(1 - \frac{1}{n}\right) \implies T(n) = O(n)$$

---

2. $T(n) = 3T\left(\frac{n}{2}\right) + cn^2, T(2) = 1$

$$T(n) = 3^2T\left(\frac{n}{2^3}\right) + 3c\left(\frac{n}{2}\right)^2 + cn^2 = \underbrace{3^3T\left(\frac{n}{2^3}\right) + 3^2c\left(\frac{n}{2^2}\right)^2 + 3c\left(\frac{n}{2}\right)^2 + cn^2}_{\text{Continued pattern}} \implies$$

$$T(n) = cn^2 + 3c\left(\frac{n}{2}\right)^2 + 3^2c\left(\frac{n}{2^2}\right)^2 + \cdots + 3^kc\left(\frac{n}{2^k}\right)^2$$

$$\frac{n}{2^k} = 1 \to n = 2^k \to k = \log_2 n$$

$$T(n) = cn^2\left(1 + 3\left(\frac{1}{2}\right)^2 + 3^2\left(\frac{1}{2^2}\right)^2 + \dots\right) = \underbrace{cn^2\left(1 + \frac{3}{4} + \frac{9}{16} + \frac{27}{64} + \dots\right)}_{a=1, r=\frac{3}{4}}$$

$$T(n) = cn^2\left(\frac{1\left(1 - \frac{3}{4}^{\log n}\right)}{1 - \frac{3}{4}}\right) = O(n^2) \implies T(n) = O(n^2)$$

Question 2

Let S be a set of two-dimensional points. Assume that all x-coordinates are distinct and all y-coordinates are distinct. A point $(x, y) \in S$ is *acceptable* if there exists a point (p, q) in S such that $x < p$ and $y < q$. Give a divide and conquer algorithm that gets a set of points as input and outputs all acceptable points. State the recurrence of your solution, and write the solution to the recurrence. You do not have to derive the recurrence.

```
 sort the array using merge sort for an O(nlogn) runtime, then:

points(arr[] S){
        if (S.size() == 1)
                return {} //empty set of acceptable points
        A = points(1..S/2)
        B = points(S/2+1..S.length)

        acceptable = {}
        index = 0

        for (x,y) in A {         //Check points from 1st half
                (p,q) = B[index] //against points from 2nd half
                if (x<p && y<q):
                        add (x,y) into acceptable
                else
                        index++
        }
```

```
        return acceptable;
}
```

$$T(n) = T\left(\frac{n}{2}\right) + O(n) \implies \underbrace{T(n) = O(n \log n)}_{\text{Master Method}}$$