COL351: Tutorial 1

Sayam Sethi

August 2021

Contents

1 Question 1

The time complexity can be formulated as $T(n) = T(\sqrt{n}) + T(n - \sqrt{n}) + c \cdot n$. On building the recursion tree, we observe that each level takes $O(n)$ steps. Thus, the time complexity can be reduced to:

$$T(n) = O(n) \cdot H(n), H(n) = 1 + H(n - \sqrt{n}) \tag{1}$$

It can be shown that $H(n) = \Theta(\sqrt{n})$ as follows:

1.1 $H(n) = o(\sqrt{n})$

In each step, the reduction in the value of n is $\leq \sqrt{n}$. Thus, the total number of steps are $\geq n/\sqrt{n} = \sqrt{n}$. Therefore, $H(n) = o(\sqrt{n})$.

1.2 $H(n) = O(\sqrt{n})$

Consider the equation $H(n) = k + O(H(n/2))$. Here, k represents the steps taken for n to reduce to a value of $n/2$. We know that the decrement of n for these k steps, lies between $\sqrt{n}$ and $\sqrt{n/2}$. Thus, $k \leq \frac{n-n/2}{\sqrt{n/2}} = \sqrt{n/2}$.

Now, reducing the above equation, we get the summation, $H(n) = \sqrt{n/2} + \sqrt{n/4} + \cdots + 1$. This summation is $O(\sqrt{n})$. Thus, $H(n) = O(n)$.

2 Question 2