CSE 321 Homework 3

Question 1:

My Equalization : (According to my algorithm)

$$T(n) = \begin{cases} 1 & \text{if } n <= 2 \\ T(n-2) + n/2 & \text{if } n > 3 \end{cases}$$

Best Case Situation:

$n=2$ condition can be best case for us. Because if we think the equalization and arr = ["Black", "White"] situation, we not need do any moves for the case. So the best case complexity is $\Omega(1)$.

Average Case and Worst Case:

If we think the equalization, average case and worst case will be same. Calculations:

$$T(n) = T(n-2) + n/2$$

$$T(n) = T(n-4) + \frac{n-2}{2} + \frac{n}{2}$$

$$T(n) = T(n-6) + \frac{n-4}{2} + \frac{n-2}{2} + \frac{n}{2}$$

$$T(n) = T(n-8) + \frac{n-6}{2} + \frac{n-4}{2} + \frac{n-2}{2} + \frac{n}{2}$$

$$T(n) = T(n-2^3) + \frac{n}{2} - 3 + \frac{n}{2} - 2 + \frac{n}{2} - 1 + \frac{n}{2}$$

$$T(n) = T(n-2^k) + (k+1).\frac{n}{2} - (k + \dots 3+2+1)$$

$$2^k = n \quad k = \log n$$

$$T(n) = T(1) + (\log n + 1).\frac{n}{2} - \frac{\log n(\log n+1)}{2}$$

$$T(n) = 1 + \frac{n.\log n + n}{2} - \frac{(\log n)^2 + \log n}{2}$$

Average case $= \Theta(n\log n)$

Worst Case $= O(n\log n)$

Question 2:

My Equalization:

$$T(n) = \begin{cases} 1 & \text{if } n <= 2 \\ 3T(\frac{n}{3}) + 3 + 2 * \Theta(\frac{n}{3}) & \text{if } n > 3 \end{cases}$$

Best Case:

If we think $n=1$ and $n=2$ situations, best case complexity will be $\Omega(1)$ to my equalization. The algorithm is entering just to the if/else cases so it will take $\Omega(1)$ time.

Average Case and Worst Case:

Average case and worst case situation will be same because my weighCoins function complexity will be same. Reason of "$\frac{n}{3}$" is thirdsOfDifference equality.

$$T(n) = 3T(\frac{n}{3}) + 3 + 2 * \frac{n}{3}$$

Master's Theorem:

$a=3$, $b=3$, $c=1$, $c \; ? \; \log_a b$

$1 = \log_3 3$ Case 2: $\Theta(n \log n)$

Average Case: $\Theta(n \log n)$

Worst Case: $O(n \log n)$

Question 3:

Insertion Sort:

$$T(n) = \begin{cases} 1 & n <= 1 \\ T(n-1) + n & n > 1 \end{cases}$$

=> I didn't use swap condition. If we think number of moves are about while loop. While loop complexity is $\Theta(n)$

$T(n) = T(n-1) + n$

$T(n) = T(n-2) + 2n$

$T(n) = T(n-3) + 3n$

$T(n) = T(n-(k-1)) + (k-1).n$ $n=k$

$T(n) = T(1) + n.(n-1)$

$T(n) = 1 + n^2 - n$ Average Case is $= \Theta(n^2)$

Quick Sort :

$$T(n) = \begin{cases} 0 & \text{if } n=1 \\ 2T(n/2) + \Theta(n) & \text{if } n>1 \end{cases}$$

I wrote partition function for choosing median number as pivot. So my quick Sort is being more faster than choosing a random number as pivot. I used swap situation two times. Swap complexity was $\Theta(n)$ to according to my partition function.

$T(n) = 2T(n/2) + n$ Masters Theorem:

$a=2,\ b=2,\ c=1$ $c \overset{?}{} \log_a b$

$1 = \log_2 2$ Case 2 $\Theta(n \log n)$

Average Case is $\Theta(n \log n)$

Expeimental Analysis

Insertion Sort (Average)

$\Theta(n^2)$

Quick Sort (Average)

$\Theta(n \log n)$

Theoretical Analysis

Insertion Sort (Average)

$\Theta(n^2)$

Qick Sort (Average)

$\Theta(n \log n)$

Question 4:

$$T(n) = \begin{cases} 0 & \text{if } n=1 \\ 2T(n/2) + 1 + \Theta(n) & \text{if } n>1 \end{cases}$$

Worst Case:

$T(n) = 2T(n/2) + 1 + n$

master's Theorem:

$a=2,\ b=2,\ c=1$ $c \overset{?}{} \log_a b$

$1 = \log_2 2$ Case 2 $\Rightarrow$ $\Theta(n \log n)$

Worst Case is $= O(n \log n)$ //

I wrote some comments for my algorithm and codes. (in files)