Problem Set 3

Due date: Electronic submission of the pdf file of this homework is due on **9/24/2021 before 11:59pm** on canvas.

Name: [redacted]

Resources. (All people, books, articles, web pages, etc. that have been consulted when producing your answers to this homework)

On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment. Furthermore, I have disclosed all resources (people, books, web sites, etc.) that have been used to prepare this homework.

Signature: [redacted]

Make sure that you describe all solutions **in your own words**.

Read chapters 2 and 4 in our textbook before attempting to solve these problems.

Problem 1 (20 points)**.** Exercise 2.3-3 on page 39 in [CLRS].

Solution. In the problem 2.3-3, we must show that T(n) is equal to $n \log n$ when n is the exact power of two, for the recurrence of

$$T(n) = \begin{cases} 2, & \text{if } n = 2 \\ 2T(n/2) + 2, & \text{if } n = 2^k, \text{ for } k > 1 \end{cases}$$

We know the general equation for the master's theorem, where T(n) = aT(n/b) + f(n). We know that $a \geq 1, b > 1$, and f(n) = $\Theta(n^k \log^p n)$. So, for 2T(n/2)+n, we know that a=2, b=2, k=1, and p=0. So, $\log_b a$ is $\log_2 2$ which is equal to 1. So, ($\log_2 2$=1) = (k=1), therefore, this falls into case 2 where, $\log_b a$ = k, so we must know what p is. p=0 so $p > -1$ thus we can use $\Theta(n^k \log^{p+1} n)$, when k=1 and p=0 so $(n^1 \log^{0+1} n) = n \log n$ hence, T(n) = $n \log n$.

Problem 2 (20 points)**.** Exercise 2.3-4 on page 39 in [CLRS].

Solution. In problem 2.3-4, we are asked to write a recurrence for the running time of the recursive version of the insertion sort.
We once again know the general form of T(n) which is T(n) = aT(n/b) + f(n). We can write a expression t(n) to represent a running there for an insertion sort. The size of the list is n, since we are given A[1... n]. So, T(n) is T(n) = T(n-1) +x(n). n-1 is determined when A[1...n-1] is used to recursively sort. x(n) determines the time taken to insert A[n] in the array A[1...n-1]. In order to place A[n] in the right position we keep moving n-1 elements where we get n' so x(n) = $\Theta(n') = \Theta(n)$.

Problem 3 (20 points)**.** Exercise 4.2-5 on page 82 in [CLRS].

Solution. In problem 4.2-5, we have to determine which method has the best asymptotic running time used in divide and conquer, and how is compares to Strassen's algorithm. We know that each of the given matrixes consists of the sizes n/68, n/70, n/72. We must use a recursive method to perform this for each matrix, and we can assume that all these values added together would produce f(n) =$\Theta(n^2)$, so the recursive relation for each matrix is...
1) 68 x 68, k=132464
so T(n) = kT(n/68) + f(n)
=kT(n/68) + $\Theta(n^2)$ (using the master's Thm)
= $\Theta(n \log_{68} k) \approx 2.795128$ (($n \log_{68} k$) comes from ($n \log_b a$))
2) 70 x 70, k=143640
so T(n) = kT(n/70) + f(n)
=kT(n/70) + $\Theta(n^2)$ (using the master's Thm)
= $\Theta(n \log_{70} k) \approx 2.795123$ (($n \log_{70} k$) comes from ($n \log_b a$))

3) 72 x 72, k=155424
so T(n) = kT(n/72) + f(n)
=kT(n/72) + $\Theta(n^2)$ (using the master's Thm)
= $\Theta(n \log_{72} k) \approx 2.795147$ (($n \log_{70} k$) comes from ($n \log_b a$))
For Strassen's, we know that $\Theta(n^{\log 7}) \approx 2.8$. The best asymptotic running time is case 2 which is 70 x 70. This is also better than the Strassen algorithm.

Problem 4 (20 points)**.** Exercise 4.2-7 on page 83 in [CLRS]

Solution. In problem 4.2-7, we must show that $a + bi$ and $c + di$, can produce ac-bd and $(ad + bc)i$ when a, b, c, and d are inputs.
So, from a, b, c, d we want the terms ac, -bd, ad, bc. To get these, ac+ad = a(c+d) because we can factor common terms, and this can be called x1. In order to achieve ac-bd we can take x1 minus some x2, so we get x1-x2 = ac - bd and solve for x2. x2= x1 +bd -ac, and x1=ac+ad, so x2 = ac +ad +bd -ac = =x2 = d(a+b), we also want some x3, where x3+x2 = ad+bc, and solve for x3. So, x3=ad+bc-ad-bd = b(c-d). Furthermore, the real part will be x1-x2 and imaginary part will be x2+x3.
To check this we can plug it in:
real: x1-x2=(ac+ad)-(ad+bd) = ac-bd
imaginary: x2+x3=(ad+bd)+(bc-bd)=ad+bc

Problem 5 (20 points)**.** Exercise 4.5-3 on page 97 in [CLRS]

Solution. For problem 5, we must use the master's theorem to show that the solution of binary search recurrence T(n) = T(n/2) + $\Theta(1)$ is $\Theta(\log n)$
We know the general formula which is T(n) = aT(n/b) +f(n). We must find a, b, and k, and p given $\Theta(n^k \log^p n)$. Based on T(n/2) + $\Theta(1)$, we know that a=1, b=2. For k and p, $\Theta(1) = \Theta(n^{\log_2 1})$, where k=0, and p=0, so $\log_b a$ = ($\log_2 1 = 0$) = (k=0), so we can use case 2, where we know p=0 so $p > -1$ so $\Theta(n^k \log^{p+1} n) = \Theta(n^0 \log^{0+1} n) = \Theta(1 \log^1 n) = \Theta(\log n)$

Work out your own solutions, unless you want to risk an honors violation!

Checklist:
☐ Did you add your name?
☐ Did you disclose all resources that you have used?
(This includes all people, books, websites, etc. that you have consulted)
☐ Did you sign that you followed the Aggie honor code?
☐ Did you solve all problems?
☐ Did you submit the pdf file of your homework? Check!