Assignment 1

Chitrarth Singh
A20387080

1.

Chitrarth Singh

Assignment 1

Question 1

Problem 2.3-3

Prove :- When n is an exact power of 2, the solution of recurrence

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

is $T(n) = n \lg n$.

Proof :-

Base case :- when $n = 2$

$T(2) = 2 \lg 2 = 2$.

By mathematical induction hypothesis

Suppose, $n = 2^y$ for $y > 1$

we assume $T(n) = T(2^y) = 2^y \lg 2^y$
$= y \cdot 2^y$.

Inductive step :.

Using the above hypothesis, the following is also true :

$n = 2^{y+1}$ for $y > 1$

$T(n) = T(2^{y+1}) = \cancel{y}\, 2^{y+1} \cdot \lg 2^{y+1}$

Assignment 1

Chitrarth Singh
A20387080

Que -1 continue:

$$= (y+1)2^{y+1}$$

$$\therefore T(2^{y+1}) = 2T\left(\frac{2^{y+1}}{2}\right) + 2^{y+1}$$

$$\Rightarrow 2T(2^y) + 2^{y+1}$$

Since, $T(2^y) = y \cdot 2^y$

$$\Rightarrow 2 \cdot y \cdot 2^y + 2^{y+1}$$

$$\Rightarrow y \cdot 2^{y+1} + 2^{y+1}$$

$$\Rightarrow 2^{y+1}(y+1)$$

$$\Rightarrow (y+1)2^{y+1}$$

So, we can conclude from the above proof that $T(n) = n \lg n$ holds true for $n = 2^k$ and $n = 2^{k+1}$ as well for $k > 1$

Assignment 1

Chitrarth Singh
A20387080

2.

Que. 2 Problem 2.3-4.

Insertion Sort :- In order to sort $A[1---n]$, we recursively sort $A[1---n-1]$ and then insert $A[n]$ into the sorted array $A[1---n-1]$.

Solution :-

The algorithm will take constant time $O(1)$ if there is only one element.

For more than one element the algorithm would recursively sort $A[1----n-1]$ items which will take $T[n-1]$ time. and to insert $A[n]$ elements into sorted Array it will take $O(n)$ time.

Therefore the recurrence relation is :-

$$T(n) = \begin{cases} O(1), & n = 1. \\ T[n-1] + O(n), & n > 1 \end{cases}$$

Assignment 1

Chitrarth Singh
A20387080

3 and 4.

Question: 3

Problem 2-3(a)

$$P(x) = \sum_{k=0}^{n} a_k x^k$$

$$= a_0 + x(a_1 + x(a_2 + \cdots + (a_{n-1} + x a_n)\cdots))$$

```
y = 0
for i = n downto 0
    y = ai + x·y.
```

Solution:-

The above code fragment takes $\Theta(n)$ time for evaluating a polynomial since there's only one FOR loop going down from n to 0.

Question 4:

Problem 3-3(a)

$\left(\frac{3}{2}\right)^n$, n^3 $\lg^2 n$ $\lg(n!)$ 2^{2^n} $n^{1/\lg n}$.

Solution:-

The exponential function with base greater than 1 grows at a faster rate than the polynomial function.

Assignment 1

Chitrarth Singh
A20387080

Que- 4 Continue

The functions according to their growth rate are:-

1. $2^{2^n} \Rightarrow \Omega(2^n)$

2. $(\frac{3}{2})^n = (1.5)^n = O(2^n)$

3. $n^3 = \Theta(n^3)$

4. $\lg(n!) = O(n \lg n)$

   Using identities we can also get

   $\lg(n!) = \Omega(\lg(2^n)) = \Omega\, n \lg 2$

   $= \Omega(n)$

5. $\lg^2 n = O(n)$

6. $n^{1/\lg n} = n^{\lg 2/\lg n}$

   $= n^{\log_n 2} = 2^{\log_n n}$

   $= 2.$

Hence growth rate of functions are:-

$2^{2^n} > (\frac{3}{2})^n > n^3 > \lg(n!) > \lg^2 n > n^{1/\lg n}$

Assignment 1

Chitrarth Singh
A20387080

5.

Question 5:

Problem 4-3 (c)

$$T(n) = 4T(n/2) + n^2\sqrt{n}.$$

Master Theorem:

$$T(n) = aT(n/b) + f(n)$$

$$\therefore \quad a = 4, \quad b = 2 \quad f(n) = n^2\sqrt{n}$$

$$\Rightarrow \frac{a f(n/b)}{f(n)} = \frac{4(n/2)^2\sqrt{n/2}}{n^2\sqrt{n}}$$

$$= \frac{\cancel{4}\cdot \cancel{n^2}/\cancel{4}\sqrt{n/2}}{\cancel{n^2}\sqrt{n}}$$

$$= \frac{\cancel{\sqrt{n}}}{\cancel{\sqrt{n}}\cdot\sqrt{2}} = \sqrt{\frac{1}{2}}$$

Since $\sqrt{\frac{1}{2}}$ is < 1, Therefore $T(n) = \Theta(f(n)) = \boxed{n^2\sqrt{n}}$

By secondary Recurrence.

$$T(n) = 4T(n/2) + n^2\sqrt{n}.$$

or

$$T(n) = 4T(n/2) + n^{2.5}$$

Assignment 1

Chitrarth Singh
A20387080

Que- 5 Continue

Lets suppose $n = 2^x$
Then the above $T(n)$ equation would become

$$T(2^x) = 4T\left(\frac{2^x}{2}\right) + 2^{2.5x}$$

$$T(2^x) = 4T(2^{x-1}) + 2^{2.5x}$$

Lets take $T(2^x) = t_x$ and $T(2^{x-1}) = t_{x-1}$
we get: $t_x = 4t_{x-1} + 2^{2.5x}$
or

$$t_x - 4t_{x-1} - 2^{2.5x} = 0 \quad -(1)$$

Applying left shift operation on above eq(1)

$$\therefore t_{x+1} - 4t_x - 2^{2.5x} = 0 \quad -(2)$$

$\therefore$ Annihilator for $t_{x+1} - 4t_x$

$$= E\langle t_x\rangle - 4\langle t_x\rangle$$
$$= (E-4)\langle t_x\rangle$$
$$= (E-4)$$

& Annihilator for $2^{2.5x} = E - 2^{2.5}$
Further we can rewrite eq (2) as

$$(E-4)(E-2^{2.5}) = 0$$

or

$$T(n) = F(x) = \alpha_1 4^x + \alpha_2 (2^{2.5})^x$$

we can neglect the lower order terms

$$\therefore F(x) = \Theta((2^{2.5})^x)$$
$$= \Theta((2^x)^{2.5})$$

Assignment 1

Chitrarth Singh
A20387080

Que- 5 continue

Since $n = 2^x$

we get, $T(n) = \Theta(n^{2.5})$

$= \Theta(n^2 \sqrt{n})$

Hence $\boxed{T(n) = \Theta(n^2 \sqrt{n})}$

Assignment 1

Chitrarth Singh
A20387080

HONESTY PLEDGE

Illinois Institute of Technology
Department of Computer Science

Honesty Pledge

CS 430 Introduction to Algorithms
Spring Semester, 2017

Fill out the information below, sign this sheet, and submit it with the first homework assignment. No homework will be accepted until the signed pledge is submitted.

I promise, *on penalty of failure of CS 430*, not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others on the homework problems in CS 430.

All work I submit will be mine and mine alone.

I understand that all resources in print or on the web, aside from the text and class notes, used in solving the homework problems *must be explicitly cited.*

Chitrarth Singh	[signature]	A20387080	18th Jan 2017.
Name (printed)	Signature	Student ID	Date