Design and Analysis of Algorithms
CS575 Spring 2023

Theory Assignment 1

Due on 2/27/2023 (Monday)

Remember to include the following statement at the start of your answers with a signature by the side. "I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment for my first offense and that I will receive a grade of "F" for the course for any additional offense."
Yash Sanjay Makwana

Please handwrite or type your answer to each question, scan or save your answers into a pdf file (with a vertical orientation so that we do not need to rotate your file to grade), and upload it to the homework submission site.

1. [20 points] Fill in all the missing values. For column A you need to compute the sums. For column B (the last two rows) you need to guess a function that does not contradict any of the yes/no answers already in the next three columns. Fill in each empty entry in the last three columns with either a yes/no answer.

Function	Function	O	Ω	θ
A	B	A = O(B)	A = Ω(B)	A = θ(B)
n^4	$n^3 \lg n$	No	Yes	No
$n\sqrt{n}$	n^2	Yes	No	No
(n + 1)!	n!	Yes	Yes	Yes
lg n	n^k where k > 0	Yes	No	No
$\sum_{i=1}^{n}(i+1) = ?$	$\sum_{i=1}^{n} i$	Yes	Yes	Yes
$\sum_{i=0}^{n-1} 3^i = ?$	$\sum_{i=0}^{n-1} 3^{i+1}$	No	Yes	No

2. [20 points] Prove the following using the original definitions of O, Ω, θ, o, *and* ω.

(a) $3n^3 + 50n^2 + 4n - 9 \in O(n^3)$

$3n^3 + 50n^2 + 4n - 9 \in O(n^3)$.

The definition of Big-Oh states that

$g(n) \in O(f(n))$: { $g(n)$: there exists positive constants c & N such that
$0 \le g(n) \le c \cdot f(n)$
for all $n \geq N$ }.

$$0 \le 3n^3 + 50n^2 + 4n - 9 \le cn^3$$

$$3n^3 + 50n^3 + 4n^3 - 9n^3 \le cn^3$$
$$57n^3 - 9n^3 \le cn^3$$
$$48n^3 \le cn^3$$

~~We see that equality holds~~

We see that inequality holds for, if we choose $c = 48$ and $N = 1$

Hence, for $n \geq 1$ the condition of Big-Oh satisfies.

Hence, $3n^3 + 50n^2 + 4n - 9 \in O(n^3)$.

(b) $1000n^3 \in \Omega(n^2)$

- $1000n^3 \in \Omega(n^2)$.

The definition of Omega states that

$g(n) \in \Omega(f(n))$: $\{g(n)$: there exists positive constants c & N such that $0 \le c \cdot f(n) \le g(n)$ for all $n \ge N\}$.

$$1000n^3 \ge cn^2$$

$$1000n^2 \ge cn$$

For $n \ge 1$ & $c \le 1000$ the condition of Omega holds ~~true~~ true. Hence $1000n^3 \in \Omega(n^2)$.

(c) $10n^3 + 7n^2 \in \omega(n^2)$

- $10n^3 + 7n^2 \in \omega(n^2)$

The definition of small omega (ω) states that

$g(n) \in \omega(f(n))$: For every positive real constant c, there ~~exists~~ exists a positive integer N, for which $g(n) \ge c \cdot f(n)$ for all $n \ge N$.

$$10n^3 + 7n^2 \ge cn^2$$

Let $c = 17$

$$10n^3 + 7n^2 \ge 17n^2$$

$$10n^3 \ge 10n^2$$

$$n \ge 1$$

Therefore, $N = 1$ for all $n \ge N$. Hence, the condition for small omega ~~hat~~ holds true.

(d) $78n^3 \in o(n^4)$

$78n^3 \in o(n^4)$.

The definition of small-oh (o) states that $g(n) \in o(f(n))$: For every positive real constant c, there exists a positive integer N, for which

$$g(n) \leq cf(n) \text{ for all } n \geq N$$

$$78n^3 \leq cn^4$$

Let $c = 78$

$$78n^3 \leq 78n^4$$

$$1 \leq n$$

$\therefore N = 1$

Hence, $78n^3 \in o(n^4)$ is proved.

(e) $n^2 + 3n - 10 \in \theta(n^2)$

$n^2 + 3n - 10 \in \Theta(n^2)$

The definition of Theta states that:
there exists positive constants c, d & N for which

$$0 \le c \cdot f(n) \le g(n) \le d f(n)$$

for all $n \ge N$.

$$c \cdot n^2 \le n^2 + 3n - 10 \le d \cdot n^2$$

First we will prove $n^2 + 3n - 10 \in O(n^2)$
Need to find constants c & N such that, for all $n \ge N$

$$n^2 + 3n - 10 \le dn^2$$

$n^2 + 3n - 10 \le n^2 + 3n^2 + 10n^2$ for all $n \ge 1$ [since n^2 is asymptotically positive].

$n^2 + 3n - 10 \le 14n^2$
for all $n \ge 1$.

If we choose $d = 4$ & $N = 1$, then
$n^2 + 3n - 10 \in O(n^2)$.

Lets prove that $n^2 + 3n - 10 \in \Omega(n^2)$

$n^2 + 3n - 10 \ge n^2$ for all $n \ge 3$ (since $3n - 10 \ge 0$ for $n \ge 3$)

$\therefore$ we can choose $c = 1$ & $N = 3$
$\therefore$ $n^2 + 3n - 10 \in \Omega(n^2)$

Since $n^2 + 3n - 10$ belong to both O (Big-Oh) & Ω (Omega).
$n^2 + 3n - 10 \in \Theta(n^2)$

3. [15 points] Prove the following using limits.

(a) $n^{1/n} \in \theta(1)$ [Hint: you can use x=$e^{\ln x}$]

- $n^{1/n} \in \Theta(1)$

To prove that $n^{1/n} \in \Theta(1)$, we need to show that there positive constants c1, c2 & N such that

$$c1 \leq \lim_{n\to\infty} n^{1/n} \leq c2 \text{ for all } n \geq N$$

Lets consider the limit:

$$\lim_{n\to\infty} n^{1/n} = \lim_{n\to\infty} e^{\ln(n^{1/n})}$$

$$= \lim_{n\to\infty} e^{\frac{1}{n}(\ln n)}$$

$$= e^{\lim_{n\to\infty} \frac{\ln(n)}{n}}$$

Using the L'Hopital's ~~rule~~ rule to evaluate the limit:

$$\lim_{n\to\infty} \frac{\ln(n)}{n} = \lim_{n\to\infty} \frac{1/n}{1} = 0$$

$$\therefore \lim_{n\to\infty} n^{1/n} = \cancel{e^1} \; e^0 = 1$$

This means that $n^{1/n}$ is bounded between 2 positive constants ~~me~~ 1 & 1 for all $n \geq 1$.

$\therefore n^{1/n} \in \Theta(1)$.

(b) $4^n \in \omega(n^k)$

- $4^n \in \omega(n^k)$

To prove that $4^n \in \omega(n^k)$ we need to show that

$$\lim_{n \to \infty} \frac{4^n}{n^k} = \infty$$

Using L'Hopital's Rule to evaluate this limit

$$\lim_{n \to \infty} \frac{4^n}{n^k} = \lim_{n \to \infty} \frac{4^n \ln 4}{k n^{k-1}}$$

$$= \lim_{n \to \infty} \frac{4^n (\ln 4)^k}{k!\, n^k} = \infty$$

Therefore, $4^n \in \omega(n^k)$.

(c) $\lg^3 n \in o(n^{0.5})$

- $\lg^3 n \in O(n^{0.5})$

$$= \lim_{n\to\infty} \frac{(\lg n)^3}{n^{0.5}}$$

$$= \lim_{n\to\infty} \frac{3(\lg n)^2 \cdot 1/n}{\frac{1}{2} n^{-1/2}}$$

$$= \lim_{n\to\infty} \sqrt{n} \cdot 6 (\lg n)^2 \cdot \frac{1}{n} = 0$$

$$(\lg n)^3 \in o(n^{0.5})$$

4. [10 points] Order the functions below by increasing growth rates (no justification required):

n^n, n, $n \ln n$, $n^{1/2}$, $2^{\lg n}$, $\ln n$, 10, $n^{1/n}$, $\sqrt{2}^{\lg n}$, $n!$, $lg(n^{10})$, 2^n

Let $g_i(n)$ be the ith function from the left after the ordering (the leftmost function has the slowest growth rate). In the order, $g_i(n)$ should satisfy $g_i(n) \in O(g_{i+1}(n))$. If two or more functions are equivalent (in terms of Θ), put them in [ ] separated by comma (e.g., [n^2, $5n^2$]).

Solution: 10, [ln(n), lg(n^{10})], $n^{1/n}$, [$\sqrt{2}^{\lg n}$, $n^{1/2}$], [$2^{\lg n}$, n] , n ln(n), 2^n, n! , n^n

5. [20 points] Let $f(n)$ and $g(n)$ be asymptotically positive functions. For each of the following conjectures, either prove it is true or provide a counter example to show it is not true.

   a. $(f(n) + g(n)) \in \Theta(\max(f(n), g(n)))$.

$- \ (f(n) + g(n)) \in \Theta(\max(f(n), g(n)))$

To prove this:

lets consider $f(n) = n^2$ & $g(n) = n$

$f(n) + g(n) = n^2 + n$

$\Theta(f(n) + g(n)) = \Theta(n^2)$

$\Theta(\max(f(n), g(n))) \cancel{=} \Theta(\max(n^2, n))$

$\cancel{= (n^2)}$

$= \Theta(n^2)$ $\therefore$ This is true

b. $f(n) \in O(g(n))$ implies $2^{f(n)} \in O(2^{g(n)})$.

$- \ f(n) \in O(g(n))$ implies $2^{f(n)} \in O(2^{g(n)})$

Consider $f(n) = 2n$ & $g(n) = n$

$f(n) \in O(g(n))$ will be $2n \in O(n)$

Need to prove

$2^{2n} \in O(2^n)$

Need to prove: $2^{2n} \leq c \cdot 2^n$

$2^n \cdot 2^n \leq c \cdot 2^n$

So

$2^n \leq c$

It will be always insufficient, even if large value of c taken for a large 'n'. The conjecture is false.

c. $f(n) \in O(g(n))$ implies $g(n) \in \Omega(f(n))$.

$f(n) \in O(g(n))$ implies $\Omega(f(n))$

If $f(n)$ is $O(g(n))$ then there exists a constant $c > 0$ & a constant N such that for all $n \geq N$, $f(n) \leq c\, g(n)$

Hence now there exists a constant $c > 0$ & a constant N such that for all $n \geq N$, $g(n) \geq \frac{1}{c} f(n)$

Hence now there exist a constant $k > 0$, namely $k = 1/c$ & a constant N such that for all $n \geq N$, $g(n) \geq k \cdot f(n)$ which is the definition of $g(n) = \Omega f(n)$.

d. $f(n) \in O(g(n))$ implies $\lg(f(n)) \in O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for sufficiently large n.

- $f(n) \in O(g(n))$ implies $\lg(f(n)) \in O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ & $f(n) \geq 1$ for sufficiently large n.

The conjecture is true.

$f(n) \in O(g(n)) \rightarrow f(n) \leq c\, g(n)$

For some constant c, this holds for all n sufficiently large (increasing c if needed, we may assume $c \geq 1$).

$\rightarrow \cancel{\lg}\ \lg(f(n)) \leq \lg(c(g(n)))$

(Since $\lg(x)$ is an increasing function)

$\rightarrow \lg(f(n)) \leq \lg(c) + \lg(g(n))$

$\rightarrow \lg(f(n)) \leq \lg(c)\lg(g(n)) + \lg(g(n))$

[Since $\lg(g(n)) \geq 1$]

$\rightarrow \lg(f(n)) \leq (\lg(c)+1)\lg(g(n))$

$\rightarrow \lg(f(n)) \in O(\lg(g(n)))$.

6. [10 points] Prove that for all integers n>0,

$$\left(\sum_{i=1}^{n} i\right)^2 = \sum_{i=1}^{n} i^3.$$

by mathematical induction. Divide your proof into the three required parts: Induction Base, Induction Hypothesis, and Induction Steps.

$$-\left(\sum_{i=1}^{n} i\right)^2 = \sum_{i=1}^{n} i^3$$

Lets take base case as $n=1$

Left Hand Side.

$$\left(\sum_{i=1}^{n} i\right)^2 = \left(\frac{n(n+1)}{2}\right)^2$$

Substituting $n=1$

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

$$= 1$$

Right Hand Side

$$\sum_{i=1}^{n} i^3 = \frac{n(n^3+1)}{2}$$

1st term = 1

Last term = n^3

Substituting $n=1$

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

The equality of LHS = RHS holds for base case.

Induction Hypothesis

Assuming the equality holds till a constant k, such that $n=k$

$$\left(\sum_{i=1}^{k} i\right)^2 = \sum_{i=1}^{k} i^3$$ is true for a constant k

lets try to prove $n = K+1$ is true.

$$LHS = \left(\sum_{i=1}^{n=K+1} i \right)^2$$

$$= \left(\sum_{i=1}^{K} i + (K+1) \right)^2$$

Squaring terms. $\left(\sum^{K} i \right)^2 = \sum_{i=1}^{K} i^3$

$$= \sum_{i=1}^{K} i^3 + \cancel{2}(K+1)\left[\frac{K(K+1)}{\cancel{2}} \right] + (K+1)^2$$

$$= \sum_{i=1}^{K} i^3 + (K+1)^2 K + (K+1)^2$$

$$= \sum_{i=1}^{K} i^3 + (K+1)^2 (K+1)$$

$$= \sum_{i=1}^{K} (K+1)^3$$

$$= \sum_{i=1}^{K+1} i^3$$

7. [15 points] Consider the following algorithm

```
for ( i = 2;  i <= n;  i++) {
    for ( j = 0;  j <= n)  {
        cout << i << j ;
        j = j + ⌊n/4⌋ ;
    }
}
```

(a) What is the output when n=4?
(b) What is the time complexity T(n). You may assume that n is divisible by 4.

a) O/P

~~2 0 2 1 2 2 2 3 2 4~~

2 0
2 1
2 2
2 3
2 4
3 0
3 1
3 2
3 3
3 4
4 0
4 1
4 2
4 3
4 4

b) Lets say that n = 24, 24 is divisible by ~~8~~ 4.

Outer loop executes n-1 time, since it starts from '2'
The outer loop runs for O(n) time.

Whatever may be the value of 'n', the inner loop runs '6' times [divisible by 4]. The inner loop run time is independent of the value of 'n', therefore run time is O(1).

Time complexity for the given algorithm

$$T(n) = O(n).$$

8. [10 points] What is the time complexity T(n) of the nested loops below? For simplicity, you may assume that n is a power of 2. That is, n = 2^k for some positive integer k. Give some justification for your answer.

```
for (i = 1; i <= n; i++){
    j = n;
    while (j >= 1){
        < body of the while loop>    //Needs Θ(1).
        j = ⌊j/2⌋;
    }
}
```

— • The outer loop runs $n+1$ times, i.e. $O(n)$ times

• The inner loop runs $\log_2 n$ times, since j is divided by 2 in each ~~that~~ iteration till it becomes 0.)

• The body of the while loop has a time complexity of $\Theta(1)$, which means it takes constant amount of time to execute.

• Therefore the total time complexity

$$T(n) = O(n\} * \log_2(n) * \Theta(1)$$
$$= O(n \log n)$$