UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

# Analysis and Design of Algorithms

**CS2102**
**Asymptotic Analysis Practice**
**2020-II**

**Duration**: 100 minutes
**Number of questions**: 5

- Attach your answers in order to keep a single PDF file. Your images should be located inside the *answers* folder and every image should have the name: *problem{1,2,3,4,5}.{pdf,png,jpg}* based on the problem you are solving. At the end there is an example of how to attach your answers in LaTeX.
- Read the questions carefully and write your answers clearly. Answers that are not legible will not have any score.
- Notes are allowed. To compile this file you can use the command latexmk -pdf main.tex

Outcomes:

a. Apply appropriate mathematical and related knowledge to computer science.

b. Analyze problems and identify the appropriate computational requirements for its solution.

---

**Problem 1 (Outcomes b) - 2 points**

The insertion sort implementation provided in the lectures uses linear search to find the position where an element should be inserted into the already sorted part of the array ($< k$). Consider that we replace the linear search with a binary search algorithm, what will be the new complexity for the worst case running time? Explain your answer

**Problem 2 (Outcomes a, b) - 8 points**

Tower of Hanoi is a mathematical puzzle that consists of 3 rods where one of them is composed by a stack of n disks arranged from largest on the bottom to smallest on top. The objective of the puzzle is to move the entire stack of disks from one rod to another given the following constraints: i) you can only move one disk at a time and ii) a larger disk can never lie above a smaller disk.

(i) Write a recursive algorithm to solve the Tower of Hanoi puzzle.

(ii) Analyze and explain the proof of correctness of your algorithm.

(iii) Sketch the recursion tree and calculate $T(n)$ based on the costs of each level.

(iv) Write the recurrence equation and solve it using induction or master method, if possible.

**Problem 3 (Outcome a) - 4 points**

Prove or disprove the following expresions:

(i) $T(n) = \Theta(T(n))$

(ii) $T(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ implies $T(n) = \Theta(h(n))$

(iii) $T(n) = \Theta(f(n))$ implies $f(n) = \Theta(T(n))$

(iv) For any 2 numbers $a, b$ one of the following **must** hold: $a < b$, $a = b$, $a > b$. Given 2 functions $T(n), f(n)$ is it true that always one of the following holds? $T(n) = \Theta(f(n))$, $T(n) = o(f(n))$, $T(n) = \omega(f(n))$

**Problem 4 (Outcome a) - 4 points**

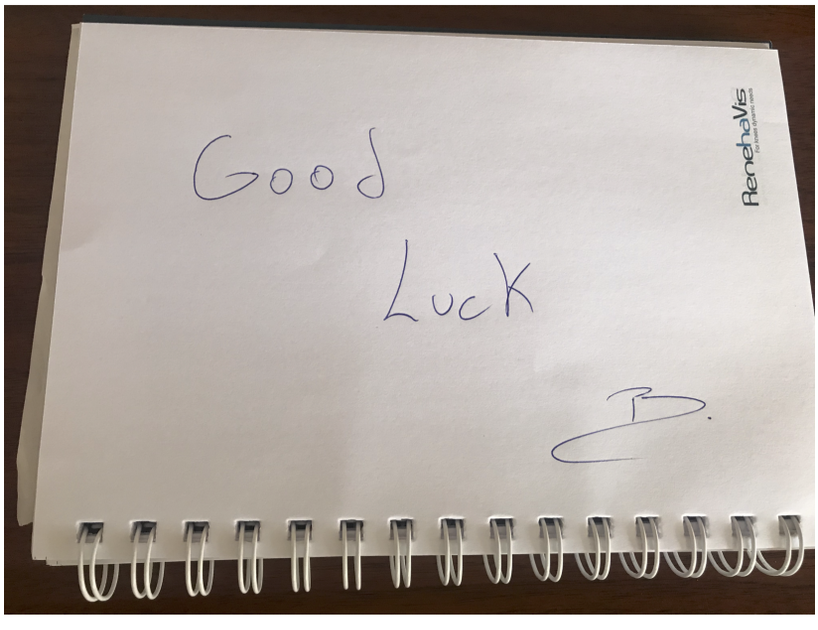
Use substitution method to solve the following expressions:

(i) $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + \Theta(n)$

(ii) $T(n) = T(n/5) + T(3n/4) + \Theta(n)$

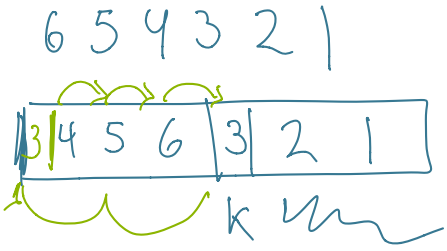
**Problem 5 (Outcome a) - 2 points**

Consider $T(n) = 4T(n/2) + f(n)$ where $f(n) = n^2 log(n)$, show if $f(n)$ is polynomically greater than $n^{log_b a}$. If that is true solve the expression using master theorem, otherwise propose an upper and lower bound to $T(n)$



**Problem 1 (Outcomes b) - 2 points**

The insertion sort implementation provided in the lectures uses linear search to find the position where an element should be inserted into the already sorted part of the array ($< k$). Consider that we replace the linear search with a binary search algorithm, what will be the new complexity for the worst case running time? Explain your answer




$\Theta(n^2)$

**Problem 3 (Outcome a) - 4 points**

Prove or disprove the following expresions:

(i) $T(n) = \Theta(T(n))$ // Reflixivity

(ii) $T(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ implies $T(n) = \Theta(h(n))$ // transitivity

(iii) $T(n) = \Theta(f(n))$ implies $f(n) = \Theta(T(n))$ // symmetry

(iv) For any 2 numbers $a, b$ one of the following **must** hold: $a < b$, $a = b$, $a > b$. Given 2 functions $T(n), f(n)$ is it true that always one of the following holds? $T(n) = \Theta(f(n))$, $T(n) = o(f(n))$, $T(n) = \omega(f(n))$

$=$ $<$ $>$

3.1) $\lim_{n\to\infty} \frac{T(n)}{T(n)} = 1, \; c>0 \Rightarrow T(n) = \Theta(T(n))$

3.2) $T(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow T(n) = \Theta(h(n))$

$\lim_{n\to\infty} \frac{T(n)}{g(n)} = c_1 > 0$ $\quad$ $\lim_{n\to\infty} \frac{g(n)}{h(n)} = c_2 > 0$

$\frac{T(n)}{g(n)} \cdot \frac{g(n)}{h(n)} \Rightarrow \boxed{\lim_{n\to\infty} \frac{T(n)}{h(n)}} \Rightarrow c_1 \cdot c_2 > 0$

$\hookrightarrow T(n) = \Theta(h(n))$ //

3.3) $T(n) = \Theta(f(n))$ implies $f(n) = \Theta(T(n))$

$\lim_{n\to\infty} \frac{T(n)}{f(n)} = c$ $\quad$ $\lim_{n\to\infty} \frac{f(n)}{T(n)} = \frac{1}{c}$ ✓

$f(n) = \Theta(T(n))$ //

3.4) $T(n) = n$

$f(n) = n^{\sin(n)+1}$ → [0 ... 1]

$f(n)$ { $n^0$ ooo $n^2$ ; 1 ooo $n^2$

$\lim_{n\to\infty} \frac{T(n)}{f(n)} = \frac{n}{n^{\sin(n)+1}}$ $\Rightarrow$ $\frac{n}{1}$ $\quad$ $\frac{n}{n^2}$

$\frac{n}{1} \to \infty$ $\quad$ $\frac{n}{n^2} \to 0$

**Problem 5 (Outcome a) - 2 points**

Consider $T(n) = 4T(n/2) + f(n)$ where $f(n) = n^2 log(n)$, show if $f(n)$ is polynomically greater than $n^{log_b a}$. If that is true solve the expression using master theorem, otherwise propose an upper and lower bound to $T(n)$

$T(n) = 4T(n/2) + n^2 \log(n)$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$\lim_{n\to\infty} \frac{n^2 \log n}{n^2} = \infty \Rightarrow f(n) = \Omega(n^{\log_b a})$

$f(n)$ is larger than $n^{\log_b c}$

$f(n) = \Omega(n^{\log_b a + \epsilon})$ ?

$\lim_{n\to\infty} \frac{n^2 \log n}{n^{2+\epsilon}} = \frac{\log n}{n^\epsilon}$ (L'Hôpital) $\frac{1/n}{\epsilon n^{\epsilon-1}}$

$\lim_{n\to\infty} \frac{1}{\epsilon n^\epsilon} = 0 = o(\ ),$ or

$f(n) \neq \Omega(n^{\log_b a + \epsilon})$

$U(n) = 4U(n/2) + n^{2+\epsilon}$ $\quad$ $n.t.\ \Theta(n^{2+\epsilon})$

$L(n) = 4L(n/2) + n^2$ $\quad$ $n.t.\ \Theta(n^2 \log n)$

(ii) $T(n) = T(n/5) + T(3n/4) + \Theta(n)$

$\sim cn$

Guess $T(n) = O(n)$

assume: $T(k) \leq ck \rightarrow k<n$

$T(n) \leq c\left(\frac{n}{5}\right) + c\left(\frac{3n}{4}\right) + cn$

$\leq \frac{4cn + 15cn + 20cn}{20}$

$\leq \frac{39cn}{20}$ ✓ $c>0$

assume $T(k) \leq 20ck$, $k<n$

$\leq \frac{20cn}{5} + \frac{20(3cn)}{4} + cn$

$\leq 4cn + 15cn + cn$

$\leq 20cn$ ✓

**Problem 2 (Outcomes a, b) - 8 points**

Tower of Hanoi is a mathematical puzzle that consists of 3 rods where one of them is composed by a stack of n disks arranged from largest on the bottom to smallest on top. The objective of the puzzle is to move the entire stack of disks from one rod to another given the following constraints: i) you can only move one disk at a time and ii) a larger disk can never lie above a smaller disk.

(i) Write a recursive algorithm to solve the Tower of Hanoi puzzle.

(ii) Analyze and explain the proof of correctness of your algorithm.

(iii) Sketch the recursion tree and calculate $T(n)$ based on the costs of each level.

(iv) Write the recurrence equation and solve it using induction or master method, if possible.

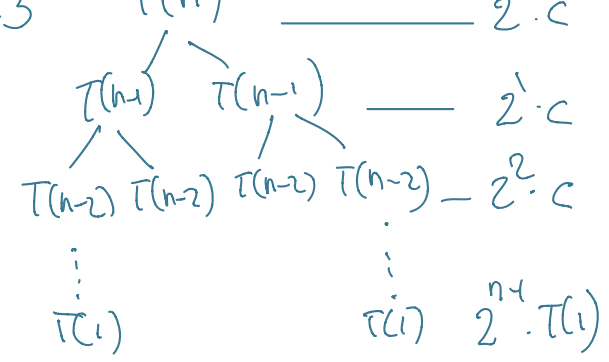
2.4) master method can't be used since doesn't met the conditions.

2.1) hanoi(n, a, b, c) — a: initial rod, b: target rod

$2T(n-1)+1$

```
if n==1:
    // move disk from a to c
    return
else:
    hanoi(n-1, a, c, b)
    // move disk from a to c
    hanoi(n-1, b, a, c)
```

2.3




# leaves $= 2^h = 2^{n-1} \Rightarrow T(n) = O(2^n)$