Introduction

CS3026 – Analysis and Design of Algorithms

Angel Napa

UTEC UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Contents

1 Efficiency Analysis

Efficiency

Example 2. *We have two computers A and B with the following characteristics and algorithms to execute.*

- *Computer A:*
  - *Speed:* 10^{10} *instructions/s,*
  - *Algorithm to execute:* INSERTIONSORT, *with complexity of* $2n^2$. *This means, with size n input, it executes* $2n^2$ *instructions.*
- *Computer B:*
  - *Speed:* 10^7 *instructions/s,*
  - *Algorithm to execute:* MERGESORT, *with complexity of* $50n \lg n$.

¿Which computer is better if …

… $n = 10^7$?

… $n = 10^8$?

Exercises

Exercise 1. *Suppose the same computer runs* InsertionSort *and* MergeSort*, where now* InsertionSort *has* $8n^2$ *complexity and* MergeSort *has* $64n\lg n$ *complexity. For what values of* n*, the algorithm* InsertionSort *is more efficient?*

Exercise 2. *What is the minimum value of n such that an algorithm with execution time of $100n^2$ is faster than another one with execution time of 2^n on the same machine?*

Exercise 3. *Make a formal proof that the* $2^n/100n^2$ *is an increasing function. Use this proof to finish the above exercise.*

2 Mathematical Tools

Logarithm properties

- Logarithms & exponentials are inverse $\log_b a = x \iff b = a^x$
- Sum and subtraction of logarithms $\log_b xy = \log_b x + \log_b y$
- Logarithm of an exponential $\log_b a^x = x \log_b a$

Applications of Induction principle

Property 1. *Show that for all positive number* n, $1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$.

Applications of Induction principle

Property 2. *For each positive integer number* n, $1+2+2^2+\cdots 2^n = 2^{n+1}-1$.

Applications of Induction principle

Property 3. *Show that, for all positive integer n,*

$$\frac{1}{2}+\frac{1}{6}+\cdots+\frac{1}{n(n+1)}=\frac{n}{n+1}$$

Applications of Induction Principle

$Show\ that\ for\ all\ positive\ integer\ number\ n \geq 1,\ \lg n \leq n.$

$Prove\ that,\ for\ all\ positive\ integer\ number\ n \geq 44,\ 8 \lg n \leq n.$

Sums

Definition 5. *Given a sequence $a_1, a_2, \ldots, a_n$ of numbers, where n is a non negative integer, we can write the sum $a_1 + a_2 + \cdots + a_n$ as*

$$\sum_{k=1}^{n} a_k.$$

If $n = 0$ then, the value of the sum is 0.

Sums

Linearity property

Given an arbitrary number c and two sequences $a_1, a_2, \ldots, a_n$ y $b_1, b_2, \ldots, b_n$, we have:

$$\sum_{k=1}^{n}(ca_k + b_k) = c\sum_{k=1}^{n} a_k + \sum_{k=1}^{n} b_k.$$

Sums

Arithmetic Series

Series where the substraction of two consecutive terms of a sequence is the same.
An example:

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Sums

Some known sums

It is known the following properties:

$$\sum_{k=0}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

$$\sum_{k=0}^{n} k^3 = \frac{n^2(n+1)^2}{4} = \left(\frac{n(n+1)}{2}\right)^2.$$

Sums

Geometric series

Series where the division of two consecutive terms of a sequence is the same.

An example:

For each real number $x \neq 1$,

$$\sum_{k=0}^{n} x^k = 1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}.$$

Sums

Harmonic Series

It is the serie

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n},$$

Sums

Proposition 3. *Show that, for all positive integer* $n \geq 1$,

$$\sum_{k=1}^{n} k \leq \frac{1}{2}(n+1)^2$$

Sums

Proposition 4. *Show that exists a constant* $c > 0$ *such that*

$$\sum_{k=0}^{n} 3^k \leq c3^n$$

holds for all positive integer n.

Sums

Proof. we will show by induction on n, that for all natural number $n \geq 0$,

$$\sum_{k=0}^{n} 3^k \leq \frac{3}{2} \cdot 3^n.$$

Sums

We can upper bound each term of the sequence. This way we can build an upper bound the serie. Let's see some examples.

$$\sum_{k=1}^{n} k \leq \sum_{k=1}^{n} n = n^2$$

Sums

Let $a_{\max} = \max_{1 \leq k \leq n} a_k$.

Then, $\sum_{k=1}^{n} a_k \leq \sum_{k=1}^{n} a_{\max} = n \cdot a_{\max}$.

Exercises

Sums

Acotaremos $\sum_{k=1}^{n} k$.

Sums

Let's bound $\sum_{k=0}^{\infty} k^2/2^k$.

Sums

Let's bound $H_n = \sum_{k=1}^{n} \frac{1}{k}$.

Binary Tree

A *binary tree* T can be defined recursively as follows.

- If T has zero nodes, then is a binary tree, else
- T is composed of 3 sets of disjoint set of nodes, a *root* node, a binary tree called its *left subtree* and a binary tree called its *right subtree*.

Complete Binary Tree

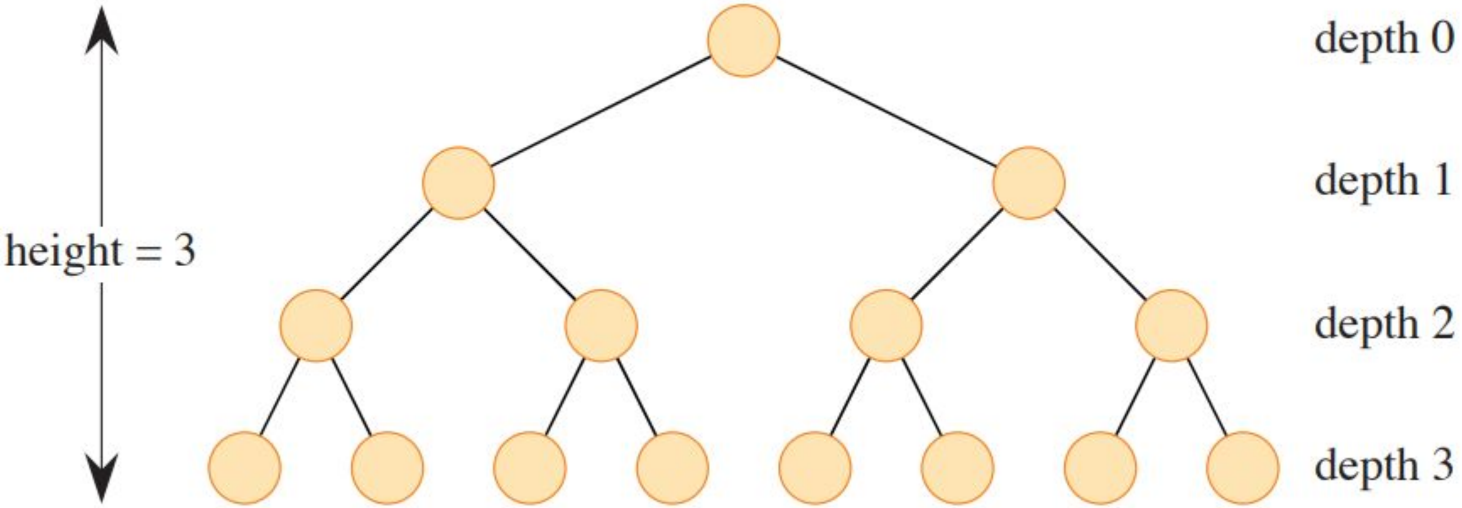

Figure B.8 A complete binary tree of height 3 with 8 leaves and 7 internal nodes.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

> Reinventa el mundo <

Thank you!

Let's keep finding mistakes!

TRANSFORMATEC