Comp #252 Notes

(Gautier) Cole Killian

January 9, 2020

Contents

§1 Lecture 01-07

§1.1 Website

http://luc.devroye.org/251.html

§1.2 Textbook

Cormen, Leisarson, Rivest, Steing, Intro to algorithms, 3rd edition.

Definition 1.1. Precise set of instructions acting on a finite input acting on a finite input and halting in finite time and using a finite amount of resources.
i.e. there is no algorithm for calculating all the digits of pi. It would be wrong to say that "my algorithm" has an infinite loop because then it would not halt in finite time.
Algorithms for everything. Need it to create a good spageti bolognese.
(world wars are usually very good for advancing technology (although we don't need another one))

§1.3 Timeline

1940: Turing and his Tape
1950: IBM Ram Model (Random access memory)
CPU constant time access very questionable
1970: Pointer based machines come into fashion
Difference: Unlimited cells. Address calcluations are not permitted / done. IBM you can go to the "next" address, but you cannot do that with the pointer based machine.

§1.4 Time / Complexity of an algorithm

Note 1.2. i, j, k, l, m, n are reserved for small f, g are for functions r, s, t are sort of grouped. t for time x, y, z o can be confused with 0 e do not use this d used for dimension

If algorithm takes finite time and finite resources, the ouput must also be finite.

$$\underbrace{x_1, \ldots, x_n}_{\text{Input}} \Rightarrow \text{Algorithm} \Rightarrow \underbrace{y_1, \ldots, y_m}_{\text{Output}}$$

Model Dependent

§1.5 Models

1. Ram Model: Every standard operationtakes 1 time unit.
2. Bit Model (competitor model cherished by computer theorists): Cost = 1 $\forall$ bit operations. (After class understand the cost of adding two binary numbers). Note that addition takes n time.
3. Oracle Model: Cost = 1 per use of the oracle.

> **Example 1.3**
> Types of oracles
> a) Comparison based oracle. Analogy is a scale.
> b) Function oracle: $f(x)$
> c) Sorting Chip. 3 inputs one output with 6 possible values (better understand why there are 6 possible values, i would have though 8)

4. Cache, communication complexity. Idea: Computers are orders of magnitudes faster than communication lines. Cost is the number of bits in an exchange between two computers. Inside cost at another computer is 0
5. Modern day computers. Memory interacts slowly with cache memory which interacts very quickly with the cpu. Cost of operation between memory and cache memory is 1. Cost of operation between cache memory and cpu is 0. (I think the memory could be thought as hard drive, and cache memory as ram)

§1.6 Time Complexity

1. Worst Case Time:

$$T_n = \max Time_n(x_1, \ldots, x_n)$$

2. Average Case Time:

$$T_n = \frac{1}{(x_1, \ldots, x_n)} \sum_{x_1, \ldots, x_n} T_n(x_1, \ldots, x_n)$$

When designing an algorithm with the user in mind the average case time is more important.

§1.7 Lambda Symbols

$$O, \Omega, \Theta, o, w, \sim$$

Let a_n, b_n be positive number sequences.

$$a_n = O(b_n) \text{ if } \exists C, n_0 : a_n \leq C \cdot b_n : (\forall n \geq n_0)$$

Example 1.4

If $T_n = O(n)$, then we are guaranteeing that T_n has no more than linear growth.
If $T_n = \Omega(2^n)$, we are guaranteeing that T_n grows at least exponentially.
If $T_n = \Theta(n^2)$, we are guaranteeing that T_n always grows at n^2. More specific information.
$a_n = o(b_n) when a_n = w(b_n) when$

$a_n = \sim (b_n) when$

§1.8 Scale

$$\underbrace{\log(n)}_{\text{polylogorithmic}(\log(n))^{\Theta}(1)}, \underbrace{\sqrt{n}, n, n\log(n), n^2}_{\text{polynomial} n^{\Theta(1)}}, \underbrace{2^n, 3^n}_{\text{Exponential}(\Theta(1))^n}, n!, n^n$$

Example 1.5 (Computing the n-th fibonacci number)

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$
$$\text{Assume } x_n \approx c^n$$
$$\Rightarrow c^n = c^{n-1} + c^{n-2}$$
$$c^2 = c + 1 \Rightarrow c = \{\frac{1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}\}$$

Solution Number 2, recursive program
Solution Number 3, dynamic programming
Solution 4, linear algebra exponentiation. Take $O(\log n)$