Efficient Algorithms and Data Structures

Ilaria Battiston

Winter Semester 2019-2020

Contents

1 Modeling issues

Modeling an algorithm is a complex task which involves measuring memory requirements, program size, power consumption but most of all computational time: comparisons, multiplications and hard disk accesses.

Measurements are taken according to theoretical analysis in a specific model of computation, giving asymptotic (lower) bounds typically focusing on the worst case.

Bounds are usually given by a function $f : \mathbb{N} : \mathbb{N}$ that maps input length to running time. The input length may be the size of the input in bits or the number of arguments, or just the number of arguments.

Performance can be measured calculating running time and storage space, or according to number of basic operations $\dashv$ easier but with less meaningful results.

1.1 Models of computation

The Touring machine is a simple model, allowing to alter the current memory location and useful to discuss computability. It is not a good model to develop efficient algorithms.

RAM has an input and an output tape, with infinite but countable number of registers holding indirectly addressed integers. Operations:

- Input operations (read);
- Output operations (write);
- Register-register transfers;
- Indirect addressing (loading content of the i-th register to the j-th);
- Branch based on comparisons;
- Jump to position x;
- Arithmetic instructions.

The cost model is uniform (every operation takes time 1) or logarithmic (depending on the content of the cells). The largest value stored in a bounded word model may not exceed 2^ω, where usually $\omega = \log_2 n$.

1.2 Complexity bounds and asymptotic notation

There are different types of complexity bounds:

- Best-case: $C_{bc}(n) := min\{C(x)|\,|x| = n\}$;
- Worst-case: $C_{wc}(n) := max\{C(x)|\,|x| = n\}$;
- Average case: $C_{avg}(n) := \frac{1}{|I_n|}\sum_{|x|=n} C(x)$
  $C_{avg}(n) := \sum_{x \in I_n} \mu(x)C(x)$.

I_n is a set of instances of length n.

There also are amortized complexity, the average cost over a worst case sequence of operations, and randomized complexity, worst case using random bits for a fixed input.

Running times are interesting for large values of n, without considering constant additive terms. A linear speed-up is always possible, but the complexity should be expressed by simple (positive) functions.

$$O(f) = \{g|\exists c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0 : [g(n) \leq c \cdot f(n)]\} \qquad g(x) \in O(f) : 0 \leq \lim_{n\to\infty} \frac{g(n)}{f(n)} < \infty$$

$$\Omega(f) = \{g|\exists c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0 : [g(n) \geq c \cdot f(n)]\} \qquad g(x) \in \Omega(f) : 0 < \lim_{n\to\infty} \frac{g(n)}{f(n)} \leq \infty$$

$$\Theta(f) = \Omega(f) \cap O(f) \qquad g(x) \in \Theta(f) : 0 < \lim_{n\to\infty} \frac{g(n)}{f(n)} < \infty$$

$$o(f) = \{g|\forall c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0 : [g(n) \leq c \cdot f(n)]\} \qquad g(x) \in o(f) : \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$

$$\omega(f) = \{g|\forall c > 0 \exists n_0 \in \mathbb{N}_0 \forall n \geq n_0 : [g(n) \geq c \cdot f(n)]\} \qquad g(x) \in \omega(f) : \lim_{n\to\infty} \frac{g(n)}{f(n)} = \infty$$

It is important to note that a function belongs to a set, does not equal to it. The constant factors can be written or ignored.

Property of linearity of sum and multiplications apply to functions with the property $\exists n_0 > 0 \forall n \geq n_0 : f(n) > 0$ for O and Θ.

Asymptotic notations must not be used within induction proofs: for any constants $a, b, \log_a n = \Theta(\log_b n)$ and the base can be ignored. In general, $\log n = \log_2 n$.

2 Recurrences

Recurrences need to be solved to bring the expression from the number of comparisons to the closed form. It can be solved in several ways:

1. Induction, proving the solution is correct;
2. Master Theorem, obtaining tight asymptotic bounds;
3. Characteristic Polynomial, to solve linear homogeneous recurrences;
4. Generating Functions, a more general technique also for non linear relations;
5. Transformation of the Recurrence to a linear one.