Chapter 1 : Complexity of a decision problem

ENSIIE - Computational complexity theory

Dimitri Watel (dimitri.watel@ensiie.fr)

2022

Decision problem

(Informal) definition

A decision problem is a problem describing inputs (*the instances*) and a question about the input for which the answer is **Yes** or **No** and depends on the input.

P, NP, NP-Complete in brief

- P: decision problems that can be solved in polynomial time;
- NP: decision problems for which a YES answer can be proved such that the proof can be verified in polynomial time;
- NP-Complete: hardest problems in NP. .

Particularly, if an NP-Complete problem is in P, then

$$\text{P} = \text{NP}.$$

which is unlikely: show that a problem is NP-Complete also show that it is probably useless to search for a polynomial algorithm to solve that problem.

Complexity of an algorithm

Definition

- The time complexity of an algorithm is the number of elementary operations performed by the algorithm before it ends.
- The space complexity of an algorithm is the quantity of memory space used by the algorithm before it ends.

Let $\mathcal{A}$ be an algorithm and $\mathcal{I}$ be an input of $\mathcal{A}$, we define $t(\mathcal{A}, \mathcal{I})$ as the time complexity of $\mathcal{A}$ when the input is $\mathcal{I}$. Similarly, $s(\mathcal{A}, \mathcal{I})$ is its space complexity.

Usually, the complexity depends on the size of the input.

Usual size of the input

Definition

The size of the input is the number of bits used to encode the input.

Example

- for a graph: number of nodes and number of edges
- for a list: number of elements
- for an integer k: $\log_2(k)$
- for a list of integers: $\sum_{k \in L} \log_2(k)$
- ...

Usually, this size is denoted by $|x|$ or n.

Definition

Let $\mathcal{A}$ be an algorithm and $\mathcal{I}_n$ be all the inputs of $\mathcal{A}$ of size n, the worst-case (time) complexity of $\mathcal{A}$ is a function f such that

$$f(n) = \max_{\mathcal{I} \in \mathcal{I}_n}(t(\mathcal{A}, \mathcal{I}))$$

We similarly define the worst-case space complexity.

Autres complexités

- Mean complexity
- Amortized complexity
- Smoothed complexity

Asymptotical worst-case time complexity

Definition

Let $\mathcal{A}$ be an algorithm and let f be its worst case complexity, we say $\mathcal{A}$

- has a worst-case complexity bounded by g asymptotically if $f(n) = O(g(n))$ when $n \to +\infty$.
- has a worst-case complexity not dominated by g asymptotically if $f(n) = \Omega(g(n))$ when $n \to +\infty$.
- has a worst-case complexity bounded both above and below by g asymptotically if $f(n) = \Theta(g(n))$ when $n \to +\infty$.

We know say, through misuse of langage, that the complexity of $\mathcal{A}$ is $O(g(n))$ if the **worst case time complexity** of $\mathcal{A}$ is asymptotically bounded by g.

Usual complexity

- Polynomial: $O(n)$, $O(n^2)$, $O(n^3)$, $O(n^c)$
- Exponential: $O(2^n)$, $O(3^n)$, $O(n!)$, $O(2^{n^2})$, $O(2^{(n^c)})$
- Constant: $O(1)$
- Linear: $O(n)$
- Quadratic: $O(n^2)$
- Cubic: $O(n^3)$
- Logarithmic: $O(\log(n))$
- Polylogarithmic: $O(\log^c(n))$
- Subexponential: $O(2^{\log^c(n)})$
- Superexponential: $O(2^{(2^n)})$

Decision problem

(Formal) definition

A decistion problem Π is a set $\mathcal{L}$ of *instances* or *inputs* and a subset $\mathcal{L}_Y \subset \mathcal{L}$ of *positive instances*. We define $\mathcal{L} \backslash \mathcal{L}_Y = \mathcal{L}_N$ the *negative instances*.

An algorithm that solve Π is able, for every instance $\mathcal{I} \in \mathcal{L}$ to decide whether $\mathcal{I} \in \mathcal{L}_Y$ or $\mathcal{I} \in \mathcal{L}_N$ in finite time.

Complexity of a decision problem

Definition

Let Π be a decision problem, then Π has a (worst-case) complexity $O(f(n))$ if there exists an algorithm $\mathcal{A}$ for Π with complexity $O(f(n))$.

The complexity class P

Definition of the class P

A decision problem Π is polynomial, or belongs to P, if its complexity is polynomial. In other words, there is a constant c such that its complexity is $O(n^c)$.

The complexity class EXPTIME

Definition of the class EXPTIME

A decision problem Π is exponential, or belongs to EXPTIME, if its complexity is exponential. In other words, there is a constant c such that its complexity is $O(2^{(n^c)})$.

$$\mathrm{P} \subsetneq \mathrm{EXPTIME}$$

The complexity class NP

(Informal) definition of the class NP

A decision problem Π is belongs to NP, if it has a polynomial verifiable proof when the answer is **YES**.

!!!

$$P \subset NP \subsetneq EXPTIME$$

Verify a solution

Definition of a verifier

Let Π be a decision problem, a verifier $\mathcal{V}$ for the **YES** answer of Π of complexity f and g is an **algorithm**

- for which the inputs are
  - an instance $\mathcal{I}$ of Π
  - a *certificate* w (a sequence of 1 and 0) of size $O(g(n))$
- that answers **YES** or **NO** in time $O(g(n))$
- that answers **YES** for at least one certificate w if the answer of $\mathcal{I}$ is **YES**
- that answers **NO** for every certificate w if the answer for $\mathcal{I}$ is **NO**

The complexity class NP

Definition of the class NP

A decision problem Π is belongs to NP if there exists a constant c and a verifier $\mathcal{V}$ for the answers **YES** with complexity $O(n^c)$ and $O(n^c)$.

!!!

$$\text{P} \subset \text{NP} \subsetneq \text{EXPTIME}$$