L3 - Notazione asintotica

Created	@September 27, 2022
Class	Alg. Strutt. Dati
Materials	

Prefazione

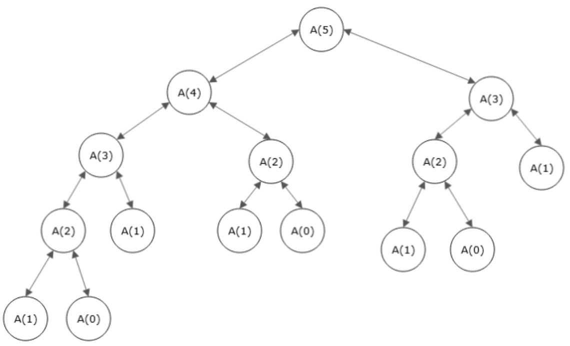
Prendiamo in esame la serie di Fibonacci:

$$x = \begin{cases} 1 \text{ se } n \in \{0,1\} \\ F(n-1) + F(n-2) \text{ altrimenti} \end{cases}$$

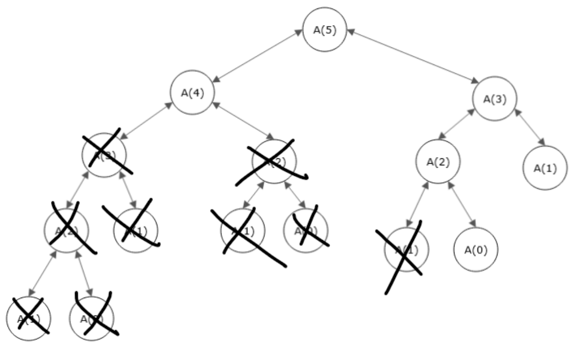
Un algoritmo per rappresentare la serie di Fibonacci può essere scritto come una funzione ricorsiva:

```
A(n):
  1) if n in {0,1}:
  2)  return 1
  3) else
  4)  return A(n-1)+A(n-2)
```

Sappiamo che questo algoritmo è esponenziale, ed è possibile stabilirlo creando un grafo ad albero, che può essere anche ridotto a una forma più semplice:



Forma completa



Forma ridotta per eliminare le ridondanze

La stessa serie può essere rappresentata con un algoritmo iterativo rappresentato da una funzione lineare:

```
A2(n):
  1) a, b <- 1
  2) while n >= 2 do:
  3)   a <- a + b
  4)   b <- a - b
  5)   n--
  6) return a
```

Notazione asintotica

Per formalizzare il concetto di algoritmo "migliore" di un altro introduciamo il concetto di Notazione Asintotica. Per ognuna di queste notazioni assumiamo le funzioni come da $\mathbb{N}$ in $\mathbb{N}$, o più estese da $\mathbb{R}$ in $\mathbb{R}$. Allo stesso modo ricordiamo che il segno "$=$" tra le funzioni è usato impropriamente; al suo posto dovrebbe essere usato il simbolo $\in$: $f(n) \in o(g(n))$

$$\text{Notazione di o-piccolo:}$$
$$f(n) = o(g(n)) \rightarrow lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$
$$\text{La notazione è la stessa della definizione di limite:}$$
$$\forall c \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} : \forall n \geq n_0 \quad f(n) \leq c * g(n)$$

$$\text{Notazione } \omega\text{-piccolo}$$
$$f(n) = \omega(g(n)) \rightarrow lim_{n \to \infty} \frac{f(n)}{g(n)} = +\infty$$
$$\text{La notazione è la stessa della definizione di limite:}$$
$$\forall M \in \mathbb{R}^+ \ \exists n_0 \in \mathbb{N} : \forall n \geq n_0 \quad f(n) \geq M$$

Per le notazioni di O, Ω e Θ espliciteremo la forma senza il limite:

$$\text{Notazione O-grande}$$
$$f(n) = O(g(n)) \rightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \ : \ \forall n \geq n_0 \quad f(n) \leq c * g(n)$$
$$\text{Notazione } \Omega\text{-grande}$$
$$f(n) = \Omega(g(n)) \rightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N} \ : \ \forall n \geq n_0 \quad f(n) \geq c * g(n)$$
$$\text{Notazione } \Theta\text{-grande}$$
$$f(n) = \Theta(g(n)) \rightarrow \exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N} : \forall n \geq n_0 \quad c_1 * g(n) \leq f(n) \leq c_2 * g(n)$$

Esaminando meglio la definizione di Θ, possiamo notare che in realtà è un'intersezione tra le due altre notazioni:

$$\Theta(g(n)) \xrightarrow{def} O(g(n)) \cap \Omega(g(n))$$

Per definizione, possiamo stabilire che le funzioni o-piccolo e $\omega\text{-piccolo}$ implicano le rispettive funzioni O-grande e $\Omega\text{-grande}$, ma NON vale il contrario.

$$f(n) = o(g(n)) \implies f(n) = O(g(n))$$
$$f(n) = \omega(g(n)) \implies f(n) = \Omega(g(n))$$

TEOREMA:

$$\text{Se } \lim_{n \to \infty} = L \in \mathbb{R}^+ \implies f(n) = \Theta(g(n))$$

DIM:

Per definizione di limite sappiamo che:

$$\forall \epsilon > 0 \;\; \exists n_0 \in \mathbb{N} \; : \; \forall n \geq n_0 \, |\frac{f(n)}{g(n)} - L| \leq \epsilon$$

Continuiamo la costruzione del limite:

$$-\epsilon \leq \frac{f(n)}{g(n)} \leq +\epsilon$$
$$\Downarrow$$
$$L - \epsilon \leq \frac{f(n)}{g(n)} \leq L + \epsilon$$

Consideriamo f definitivamente crescente e moltiplichiamo tutto per $g(n)$:

$$\forall n \geq n_0 \; g(n) > 0$$
$$\Downarrow$$
$$g(n) * (L - \epsilon) \leq f(n) \leq g(n) * (L + \epsilon)$$

Questa è esattamente le definizione di Θ. Abbiamo quindi dimostrato il teorema