07. Complessità Computazionale

Corso di Algoritmi e Linguaggi di Programmazione Python/C

Angelo Cardellicchio
angelo.cardellicchio@stiima.cnr.it

Outline

- Analisi degli Algoritmi
- Complessità degli Algoritmi
  - Complessità Spaziale
  - Complessità Temporale
- Complessità di caso peggiore
- Esercizi

Analisi degli Algoritmi

- Analisi **a priori**
  - *Teorica*
  - *Assunzioni forti*
  - *Deterministica*
- Analisi **a posteriori**
  - *Empirica*
  - *Dipende dal contesto*
  - *Beneficia di un'analisi statistica*

Complessità degli Algoritmi

- **Ipotesi:**
  - algoritmo X
  - parametri di implementazione C (es. linguaggio di programmazione, numero di input, etc.)
- Complessità legata a:
  - Numero di istruzioni
  - Spazio occupato in memoria durante l'esecuzione
  - Implementazione (*la supporremo ottimale!*)

Complessità Spaziale (1)

- La **complessità spaziale** $C_S(X)$ è data da due contributi:
  - una parte fissa $C_{S_F}(C)$;
  - una parte variabile $C_V(C)$.
- Risulta quindi:

$$C_S(X) = C_{S_F}(C) + C_{S_V}(C)$$

Complessità Spaziale (2)

- Ad esempio:

```
SUM_ONE(P, Q)
Step 1 -> START
Step 2 -> R = P + Q + 1
Step 3 -> STOP
```

- Analisi a priori

$$C_{S_{PR}}(X) = 1 + 3$$

- Analisi a posteriori
  - *Ipotesi: un intero pesa m_{int} bit, con $m_{int} = 32$*

$$C_{S_{PO}}(X) = 4 \cdot m_{int} = 128\ bit$$

Complessità Temporale

- La **complessità temporale** $T(I)$ tiene conto del numero di operazioni effettuate.

```
SUM_ONE(P, Q)
Step 1 -> START
Step 2 -> R = P + Q + 1
Step 3 -> STOP
```

- Facciamo due operazioni di addizione, più un'assegnazione.
- **Analisi a priori:** $C_{T_{PR}}(X) = 3$
- **Analisi a posteriori:** processore ad 1 Hz: $C_{T_{PO}}(X) = 4s$

Complessità di caso peggiore (1)

- Vogliamo conoscere il *limite massimo* in termini di tempo o spazio per X
- Usiamo la *notazione O-grande*, o *Big-O notation*
- Ad esempio:

$$C_T(X) = O(n)$$

- indica che la complessità temporale dell'algoritmo X è nell'ordine di n
  - n è di solito un parametro come numero di input, numero di cicli necessari, etc.
  - La notazione precedente implica che nel caso peggiore saranno necessarie n operazioni per risolvere l'algoritmo X.

Complessità di caso peggiore (2)

- **Ci sono due regole per determinare la complessità di caso peggiore per una funzione** $f(x)$
- Se $f(x)$ è una somma di più termini, si **considera** solo quello con il **tasso di crescita maggiore** (ovvero quello con l'esponente più elevato).

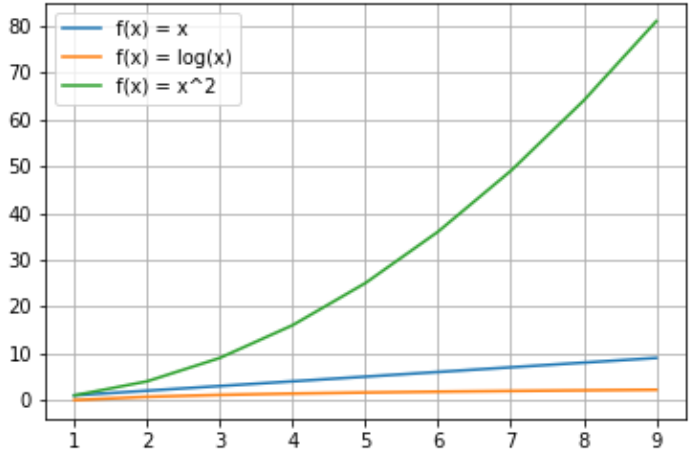
$$f(x) = x^2 + x \Rightarrow C_T\big(f(x)\big) = O(x^2)$$

- Se $f(x)$ è un prodotto di più fattori, si possono **omettere** quelli **costanti**.

$$f(x) = 3 \cdot x^2 \Rightarrow C_T\big(f(x)\big) = O(x^2)$$

Complessità di caso peggiore (3)

- Come regola generale, le funzioni di tipo polinomiale crescono più lentamente delle potenze, e più velocemente delle funzioni logaritmiche.
- Le funzioni di tipo esponenziale tendono ad infinito più velocemente di tutte le altre.



Esercizi

1. Calcolare la complessità di caso peggiore di $f(n) = 2 + n^2 \cdot n^3$.
2. Calcolare la complessità di caso peggiore di $f(n) = \mathrm{n}^2 + \log(\mathrm{n})$.
3. Calcolare la complessità di caso peggiore di un ciclo for con n iterazioni in ciascuna delle quali viene effettuata un'addizione.
4. Calcolare la complessità di caso peggiore di due cicli for annidati, ciascuno con n iterazioni, ed all'interno dei quali sono sommati i valori del contatore di ciascuno dei cicli.

Soluzioni (1)

1. La complessità di caso peggiore è data dal prodotto degli esponenti di n^2 ed n^3, per cui vale $O(n^5)$.
2. La complessità di caso peggiore è data dal termine che cresce maggiormente, ovvero n^2, quindi è un $O(n^2)$.
3. Lo pseudocodice per un ciclo for è:

```
n = 10;
for i da 1 a n:
    incrementa i;
endfor
```

In questo caso, ci sono solo dieci operazioni, per cui la complessità è un $O(n)$.

Soluzioni (2)

4. Lo pseudocodice è:

```
n = 10;
for i da 1 a n:
    for j da 1 a n:
        scrivi i;
        scrivi j;
    endfor
endfor
```

In questo caso, ci sono solo dieci iterazioni interne e dieci esterne, per cui la complessità è un $O(n^2)$.

Domande?

42