Appunti di Algoritmi e Strutture dati
A.A. 2022/2023

1 Fondamenti

1.1 Algoritmi e loro rappresentazione

Un **problema** é un quesito che richiede la determinazione o la costruzione di uno o piú enti matematici che soddisfino le condizioni specificate nell'enunciato. Con *problema* si denota l'enunciato generale, con *istanza* si denota un caso particolare del problema, ovvero un insieme specifico di dati per il quale si vuole ottenere una soluzione.

Un **algoritmo** é una sequenza di azioni non ambigue che risolve un problema utilizzando un insieme di azioni elementari, eseguibili da un opportuno esecutore. Con **programma** si denota la rappresentazione di un algoritmo utilizzando un linguaggio (con opportune "traduzioni") direttamente comprensibile da un elaboratore.
L'algoritmo é specificato da un insieme ben definito di dati in input e in output, deve essere eseguibile in un numero finito di passi, fornire il risultato corretto per ogni possibile input ed essere abbastanza generale da essere applicabile a un'intera classe di problemi.

Lo **pseudocodice** é un linguaggio astratto ed informale, inteso per uso umano, utilizzato per descrivere un algoritmo. Utilizza la struttura di un linguaggio di programmazione normale ma non é vincolato nella sintassi ed é integrabile con linguaggio naturale o notazioni matematiche compatte.

1.2 Confronto di algoritmi

Nel confrontare gli algoritmi, ci si astrae da tutti gli aspetti dipendenti dall'implementazione.
La risorsa principale su cui ci si basa é il **tempo di esecuzione**, quantificato non in secondo ma in accessi alla RAM.
L'algoritmo viene visto come una funzione $f(n)$ (dove n é l'input fornito) di cui si considera solamente il termine dominante (solitamente si trascura anche il coefficiente). Per confrontare due algoritmi, si confrontano le due rispettive funzioni per $n \to \infty$.
La crescita *asintotica* delle funzioni viene descritta con la notazione O grande

(ordine di grandezza). L'**ordine di grandezza** é la piú piccola funzione maggiorante (siccome ne esistono infinite).

Definizione. Siano $f : R \to R, g : R \to R$, diciamo che $f(x)$ é $O(g(x))$ se $\exists C, K : \forall x > K$ si ha $f(x) \leq C \cdot g(x)$.

Esempi

- Se si vuole dimostrare $f(x) = x^2 + 2x + 1$ é $O(x^2) \to x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2 \iff 3x^2 - 2x - 1 \geq 0 \iff x \geq 1$ ($\frac{2 \pm \sqrt{16}}{6} \to \frac{2+4}{6} = 1$). Ho dimostrato che la definizione vale per $K = 1, C = 4$.

- Per dimostrare invece che $f(n) = \frac{n(n+1)}{2}$ <u>non é</u> $O(n) \to \frac{1}{2}(n^2 + n) \leq Cn \iff n^2 + n \leq 2Cn \iff n^2 \leq n(2C - 1) \iff n \leq 2C - 1$. Essendo C costante, $\forall C \in R, \exists n \in R : n > 2C - 1$, quindi $f(n)$ <u>non é</u> $O(n)$.

Alcuni ordini di grandezza ben noti in ordine crescente: **costante** ($O(1)$), **logaritmico** ($O(\log n)$), **lineare** ($O(n)$), **log lineare** ($O(n \log n)$), **polinomiale** ($O(n^k)$ con k costante), **esponenziale** ($O(C^n)$ con C costante).

1.2.1 Analisi dei casi

Nell'analisi degli algoritmi si possono analizzare 3 casi: caso ottimo (migliore possibile), medio e pessimo (peggiore possibile). Principalmente si analizza il caso pessimo e il caso medio, anche se l'analisi di quest'ultimo é spesso molto piú complicata delle altre due perché richiede un'analisi statistica.

Esempio: Ricerca sequenziale.
Problema: dato un array v e un valore x, restituire l'indice della prima occorrenza di x in v, o -1 se x non é presente.

1. Caso ottimo: l'elemento é all'inizio della lista ($O(1)$)
2. Caso pessimo: l'elemento é in fondo o non é presente, quindi si itera su tutti gli elementi ($O(n)$)
3. Caso medio: per ipotesi, l'elemento é sempre presente e la probabilitá p_i che l'elemento si trovi alla posizione i sia la stessa per ogni i, quindi $p_i = \frac{1}{n}$. Se l'elemento é nella prima posizione, devo controllare una sola volta, nella seconda due volte, nella terza tre volte e cosí via fino a n: questo lo posso esprimere con la somma dei primi n numeri, ovvero con la serie geometrica $\displaystyle\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Moltiplico il tutto per la probabilitá p_i, che é la stessa per ogni elemento e ottengo $\displaystyle\frac{1}{n} \cdot \frac{n(n+1)}{2} = \frac{1}{2}(n+1)$, ovvero $O(n)$.