

Algoritmi e Strutture Dati

Introduzione al Corso

Maria Rita Di Berardini (Camerino), Emanuela Merelli (Ascoli)¹

¹Scuola di Scienze e Tecnologie - Sezione di Informatica
Università di Camerino

Parte I

Il concetto di Algoritmo

Il concetto di “algoritmo”

Informalmente, un algoritmo è una procedura computazionale ben definita eseguita per risolvere un dato problema computazionale

È una sequenza di passi computazionali che prende dei valori come input e produce altri valori come output

Più precisamente, un algoritmo è un procedimento di calcolo esplicito, descrivibile con un **numero finito di regole** che conduce al risultato dopo un **numero finito di operazioni**, cioè di applicazioni delle regole

Un esempio

Definizione del problema: ricerca del minimo in un array:

$$\min(A[1 \dots n]) = a \text{ sse } a \leq A[i] \text{ per ogni } i = 1, \dots, n$$

(stabilisce una relazione tra input e output)

Algoritmo (descrive una procedura computazionale per realizzare tale relazione):

```

min(A)                                min(A) //se A è ordinato
  a ← A[1]                              return A[1]
  for i ← 2 to length[A]
    do if A[i] < a
      a ← A[i]
  return a

```

Un pò di storia

Etimologia:

Il termine **algoritmo** significa procedimento di calcolo

Deriva dal termine latino medievale **algorismus**, che a sua volta deriva dal nome del matematico usbeco *Abu Jafar Mohammad ibn-Musa al-Khowarismi*, vissuto nel IX (?) secolo

Algoritmi nella storia

- Algoritmi di tipo numerico sono stati studiati da babilonesi e indiani
- Algoritmi in uso ancora oggi sono stati studiati da matematici greci 2000 anni fa (Algoritmo di Euclide per il MCD, algoritmi geometrici, ...)



Un pò di storia. Dal 1930

La teoria degli algoritmi ha iniziato a stabilizzarsi agli inizi del XX secolo, mentre ...

- 1928-1930 - David Hilbert - matematico tedesco, formalizza il concetto di algoritmo per definire il problema della “decisione”: *si può dimostrare che l'insieme degli assiomi dell'aritmetica è consistente?*
- 1930 - Kurt Gödel, risponde “NO”. Il Teorema di incompletezza di Gödel stabilisce infatti che ogni sistema formale può provare la propria consistenza se e solo se il sistema stesso inconsistente.

Il primo teorema di incompletezza di Gödel dimostra che qualsiasi sistema che permette di definire i numeri naturali necessariamente incompleto: esso contiene affermazioni di cui non si pu dimostrare n la verit n la falsità.

Teorema di Kurt Gödel - 1930

Il teorema di Kurt Gödel, ha un'altra interpretazione esprimibile nel contesto informatico.

Nella logica del primo ordine l'insieme di tutti i teoremi di una teoria è un insieme ricorsivamente enumerabile: ciò significa che è possibile scrivere un **algoritmo** che possa generare, prima o poi, ogni dimostrazione valida.

È possibile scrivere un programma per computer in grado di determinare con certezza se una certa affermazione vera o falsa? Il teorema di Gödel dice che ciò, in generale, non è possibile.

Macchina di Turing - 1936

Uno dei più celebri modelli matematici che definiscono il concetto di algoritmo è la macchina di Turing.

Essa rappresenta una sorta di computer ideale corredato di un programma da eseguire.

Perchè parliamo di algoritmi

Le tecniche di progettazione di algoritmi e di analisi di correttezza e di efficienza si sono evolute nella seconda metà del XX secolo grazie alla diffusione dei calcolatori elettronici

Ovunque si impieghi un calcolatore occorrono algoritmi corretti e efficienti che ne utilizzino al massimo le possibilità. Esempi di algoritmi efficienti:

- controllo dei voli aerei
- regolazione reattori nucleari
- reperimento d'informazioni da archivi
- smistamento di comunicazioni telefoniche
- gioco degli scacchi
- controllo della produzione di una catena di montaggio

Come valutiamo gli algoritmi

Risolve correttamente il problema?

- un algoritmo si dice corretto se, per ogni istanza di input, si ferma con l'output corretto
- un algoritmo corretto risolve il problema computazionale dato
- dimostrazione matematica, descrizione informale

Risolve il problema in maniera efficiente (analisi di algoritmi)?

- definizione di **efficienza** (tempo o memoria)
- alcuni problemi non possono essere risolti in maniera efficiente
- esistono delle soluzioni ottime: non è possibile fare di meglio

Algoritmi e Programmi

Gli algoritmi vengono descritti tramite **programmi**, che si avvalgono di istruzioni e costrutti dei **linguaggi di programmazione** per essere eseguiti da calcolatori elettronici

I programmi sono formulazioni **concrete** di algoritmi **astratti** che si basano su particolari rappresentazioni dei dati, e utilizzano operazioni di manipolazione dei dati, messe a disposizione da uno specifico **linguaggio di programmazione**

Le proprietà degli algoritmi sono talmente **fondamentali**, **generali** e **robuste**, da essere indipendenti dalle caratteristiche di specifici linguaggi di programmazione o di particolari calcolatori elettronici

Strutture Dati

Il concetto di algoritmo è inscindibile da quello di **dato**: per risolvere un problema computazionale, occorre organizzare ed elaborare dati

Un algoritmo può essere visto come un manipolatore di dati: a fronte di dati in ingresso che descrivono il problema producono dati in uscita come risultato del problema

È fondamentale che i dati siano ben organizzati e **strutturati** in modo che il calcolatore li possa elaborare efficientemente

“Clever” e “Efficient”

Obiettivo:

Studiare i modi più appropriati di organizzare i dati di un problema al fine di realizzare un algoritmo efficiente

Domanda:

- Che cosa intendiamo per **appropriato** “clever”?
- Che cosa intendiamo per **efficiente** “efficient”?

“Clever” e “Efficient”

Data Structure

Algorithms

Liste, pile, code

Insert

Heaps

Delete

Alberi binari di ricerca

Find

B-trees

Merge

Tabelle Hash

Shortest path

Grafi