

Sul concetto di algoritmo

Alcuni esempi di istruzioni in sequenza

Che cos'è un algoritmo e sue caratteristiche

A cura di Bruno Pizzamei

(A) Una ricetta di cucina: spaghetti cacio e pepe

1. Procurate gli ingredienti: *Spaghetti 320 g, Pecorino romano stagionatura media, da grattugiare 200 g, Pepe nero in grani 5 g*
2. mettete sul fuoco una pentola con l'acqua per cuocere la pasta: riempitela con circa metà dell'acqua che usate di solito, in questo modo sarà più ricca di amido.
3. Nel frattempo grattugiate il Pecorino romano e trasferitelo quasi tutto in una ciotola, tenendone un po' da parte per l'impiattamento.
4. Quando l'acqua sarà arrivata a bollore, salate moderatamente e tuffate gli spaghetti.
5. pestate i grani di pepe con un batticarne (oppure usate un macinino lasciandolo piuttosto lento in modo da avere anche pezzetti più grandi di pepe) Versate il pepe in un'ampia padella antiaderente e tostatelo a fuoco dolce, mescolando con un mestolo di legno. Sfumate con un paio di mestoli dell'acqua di cottura della pasta Continuate a mescolare a fuoco dolce.
6. Scolate gli spaghetti 2-3 minuti prima rispetto al tempo indicato sulla confezione e trasferiteli nella padella, avendo cura di conservare l'acqua di cottura
7. Terminate la cottura degli spaghetti in padella, risottandoli, aggiungendo quindi acqua calda al bisogno e rimestando continuamente con le pinze.
8. Aspettate che il fondo della padella sia quasi asciutto prima di aggiungere altra acqua. potete preparare la crema di Pecorino (non iniziate prima perché tenderebbe a rapprendersi o a risultare troppo fredda rispetto alla temperatura della pasta): versate un mestolo di acqua di cottura calda nella ciotola con il Pecorino grattugiato poi mescolate energicamente con una frusta a mano.
9. La consistenza che dovrete ottenere è più pastosa che cremosa, non preoccupatevi se sarà molto asciutta. Qualora una volta pronta la crema la pasta non dovesse essere pronta per la mantecatura, ponete brevemente la ciotola sulla pentola

con l'acqua calda continuando a mescolare con la frusta, così da avere la crema a una temperatura simile a quella della pasta

10. Quando gli spaghetti saranno giunti a cottura, spegnete il fuoco e unite la crema di Pecorino rimestando di continuo con le pinze.

11. Per la cremosità ideale regolatevi aggiungendo più Pecorino oppure poca acqua di cottura al bisogno. Servite i vostri spaghetti cacio e pepe spolverizzando con il restante Pecorino grattugiato e altro pepe macinato al momento a piacere.

(B) Come fare la riga di base all'uncinetto di una sciarpa.

1 Procurati i materiali.

Per un principiante dell'uncinetto, è importante cominciare con i materiali giusti. La tua sciarpa dovrebbe essere sui 150 cm, quindi avrai bisogno di almeno 180 cm di filato ritorto, un uncinetto da 8 e un ago da rammendo per finire la sciarpa alla fine. Avrai anche bisogno delle forbici per tagliare la sciarpa dal filo alla fine.

2 Impugna l'uncinetto correttamente.

Benché il modo d'impugnare l'uncinetto varia da persona a persona, di solito si tiene in mano come una penna. Il pollice e l'indice dovrebbero stare sulla parte piatta dell'uncinetto, mentre il resto delle tue dita rimangono dietro. Tieni sempre l'uncinetto nella mano destra, anche se sei mancino. Quindi userai la mano sinistra per fare un anello con il filo e tenere la tua creazione.

3 Fai il primo anello.

Per creare l'inizio della tua riga di base, dovrai fare un nodo scorsoio e farlo scivolare sull'uncinetto. Per farlo:

-Avvolgi la fine del filo (qualche cm dalla fine) attorno all'indice della mano sinistra in un anello.

-In cima al dito/anello devi infilare la parte centrale del filo in un secondo anello attraverso il primo.

-Togli il dito e infila il primo anello che ti sei appena sfilato per tirarlo e stringere il nodo.

-Metti il nodo alla fine dell'uncinetto e sistemalo tirando i fili per stringere o allentare.

4 Aggiungi la prima catenella.

Per fare una catenella tieni il filo sciolto nella mano sinistra e usa l'uncinetto nella mano destra per formarlo.

-Tieni il filo tirato, e fai avvolgere l'uncinetto sotto al filo.

-Aggancia la sezione di filo avvolta attorno all'uncinetto (chiamata filo gettato).

-Tira il filo che hai agganciato attraverso il primo anello.

5 Lavora la riga di base.

Usa lo stesso schema che hai usato per fare la prima catenella per fare tutta la riga di base. La riga di base dovrebbe essere larga circa 12-18 catenelle, a seconda di quanto tu voglia larga la tua sciarpa. Per aggiungere altre catenelle ripeti la stessa operazione: tieni il filo tirato, avvolgi l'uncinetto sotto al filo per creare un anello, aggancia il filo e tiralo attraverso l'anello che è già pronto sull'uncinetto. Tutte le volte che fai questo, crei una catenella.

(C) Problema: dato un mazzo di chiavi, abbiamo la necessità di trovare, all'interno di questo, la chiave che apre un determinato lucchetto.

Vediamo tutti i passi della sequenza:

1. Si sceglie una chiave dal mazzo e la si marca con un pennarello

2. Si tenta di aprire il lucchetto con la chiave appena marcata; se funziona, si va al passo 4;
3. Se non funziona, si va avanti a controllare la chiave successiva e:
 - Se non è marcata, la si marca e si torna al passo 2
 - Se verificate tutte le chiavi marcate nessuna è valida, si notifica che nel mazzo non è presente la chiave che apre il lucchetto
4. Fine della ricerca.

(D) Calcolo della moltiplicazione 4.231 x 97

$$\begin{array}{r}
 \begin{array}{cccc} \boxed{4} & \boxed{2} & \boxed{3} & \boxed{1} \end{array} \times \\
 \begin{array}{cc} \boxed{9} & \boxed{7} \end{array} = \\
 \hline
 \begin{array}{ccccc} \boxed{2} & \boxed{9} & \boxed{6} & \boxed{1} & \boxed{7} \end{array} + & \text{Prodotto primo fattore per} \\
 & \text{l'unità del secondo fattore} \\
 \begin{array}{cccccc} \boxed{3} & \boxed{8} & \boxed{0} & \boxed{7} & \boxed{9} & \boxed{0} \end{array} = & \text{Prodotto primo fattore per la} \\
 & \text{decina del secondo fattore} \\
 \hline
 \begin{array}{cccccc} \boxed{4} & \boxed{1} & \boxed{0} & \boxed{4} & \boxed{0} & \boxed{7} \end{array} & \text{Prodotto totale dato dalla} \\
 & \text{somma sei singoli prodotti}
 \end{array}$$

Moltiplico la prima cifra a destra del secondo fattore per tutto il primo fattore, il 7 moltiplicherà tutte le cifre del primo fattore partendo da destra (ricorda di sommare i riporti);

1. scrivo il risultato sotto (*nei quadrati rossi*);
2. moltiplico la seconda cifra del secondo fattore per tutto il primo fattore;
3. scrivo il risultato sotto quello precedente, ma spostato di una cifra verso sinistra che occupo con uno zero (*nei quadrati azzurri*);
4. ripeto come sopra (*se il secondo fattore fosse formato da più di due cifre*) ma ad ogni cifra aumento gli spostamenti di una posizione verso sinistra.
5. Ora le unità sono tutte sulla stessa colonna, come pure le decine, le centinaia, ecc., e le sommiamo esattamente come si fa in un'addizione.

L' algoritmo

Nelle righe precedenti abbiamo preso in considerazione delle attività che dovevano essere svolte seguendo una serie di istruzioni.

Cerchiamo ora di chiarire cosa significa **algoritmo**.

Quali sono le caratteristiche di un algoritmo? In che modo questo concetto è vitale nell'informatica?

Algoritmo: una definizione

Per algoritmo si intende ***un procedimento che attraverso una sequenza finita di istruzioni o operazioni che risolve un problema o esegue un compito specifico sui dati per ottenere i risultati.***

In altre parole, è una procedura ben definita che prende un input, esegue una serie di operazioni e produce un output.

In un algoritmo, l'input e l'output sono due concetti fondamentali:

Input: Sono i dati iniziali forniti all'algoritmo. Questi dati possono essere numeri, stringhe, immagini o qualsiasi altra forma di informazione che l'algoritmo deve elaborare.

Ad esempio, in un algoritmo che calcola la somma di due numeri, gli input sarebbero i due numeri da sommare.

Output: È il risultato finale prodotto dall'algoritmo dopo aver elaborato gli input.

Continuando con l'esempio precedente, l'output dell'algoritmo che calcola la somma di due numeri sarebbe il valore della somma.

Gli algoritmi sono fondamentali in informatica e matematica, ma possono essere applicati, come abbiamo, visto in molti altri campi, come la cucina (ricette), la gestione aziendale (processi) e la vita quotidiana (routine).

Gli algoritmi possono essere rappresentati in vari modi, come diagrammi di flusso, pseudocodice o linguaggi di programmazione.

Il **pseudocodice** è un modo per descrivere un algoritmo utilizzando una combinazione di linguaggio naturale e strutture di programmazione. Non segue una sintassi rigorosa come i linguaggi di programmazione veri e propri, ma è abbastanza dettagliato da permettere a un programmatore di tradurlo facilmente in codice.

Pseudocodice per un algoritmo che calcola la somma di due numeri:	Pseudocodice per un algoritmo che trova il maggiore di tre numeri
<pre> Inizio Leggi il primo numero e memorizzalo in A Leggi il secondo numero e memorizzalo in B Calcola la somma di A e B e memorizzala in C Stampa il valore di C Fine </pre>	<pre> Inizio Leggi il primo numero e memorizzalo in A Leggi il secondo numero e memorizzalo in B Leggi il terzo numero e memorizzalo in C Se A è maggiore di B e A è maggiore di C allora Il maggiore è A Altrimenti se B è maggiore di A e B è maggiore di C allora Il maggiore è B Altrimenti Il maggiore è C Stampa il valore del maggiore Fine </pre>

Un **linguaggio di programmazione** è un sistema formale di notazione che permette di scrivere programmi che possono essere eseguiti da un computer. I linguaggi di programmazione forniscono una serie di istruzioni e regole sintattiche che consentono ai programmatori di comunicare con i computer e di creare software.

Programma e algoritmo

Un **programma** è un insieme di istruzioni che un computer esegue per svolgere una determinata funzione.

Un'**app** (abbreviazione di applicazione) è un tipo di programma progettato per eseguire compiti specifici per l'utente finale, spesso su dispositivi mobili come smartphone e tablet.

Quindi, mentre tutte le app sono programmi, non tutti i programmi sono app.

Un algoritmo è un insieme di regole o istruzioni passo-passo per risolvere un problema o eseguire un compito.

In altre parole un programma è l'esecuzione pratica di uno o più algoritmi.

Un po' di storia

La parola algoritmo deriva dal nome del matematico **Mohammed ibn-Musa al-Khwarizmi**, che faceva parte della corte reale di Baghdad e che visse tra il 780 e l'850 circa.

Questo matematico viene considerato tra i primi ad aver fatto accenno a questo concetto, con la scrittura del libro ***"Regole di ripristino e riduzione"***.

Il termine indicò nel medioevo i procedimenti di calcolo numerico utilizzando le cifre indo-arabiche. Anche la parola algebra deriva da questo matematico.

Il **concetto di algoritmo ha origini antichissime**: se ne trovano tracce in documenti conosciuti come i papiri di Ahmes, che contengono una collezione di problemi con relativa soluzione comprendendo un problema di moltiplicazione che lo scrittore dice di aver ripreso da altri papiri anteriori di 12 secoli.

Algoritmi in informatica

Ora che abbiamo visto cos'è un algoritmo, è bene chiarirne i suoi campi di applicazione, che non riguardano solo la matematica.

Gli algoritmi sono ampiamente utilizzati in tutte le aree dell'**IT** (Information Technologies).

Volendo fare un esempio di algoritmo in informatica, i motori di ricerca come Google sono basati proprio su questo concetto per poter rispondere quanto più coerentemente alla richiesta di un utente.

Caratteristiche di un algoritmo

Riassumiamo alcune caratteristiche e proprietà fondamentali legate a questo concetto di algoritmo.

Esistono diverse **tipologie di algoritmo**, ma possiamo individuare alcune **proprietà comuni**, senza le quali un algoritmo non potrebbe essere definito come tale:

Un algoritmo deve essere:

1. **finito**, è costituito da un numero finito di istruzioni e presenta una fine;
2. **deterministico**, per ogni istruzione esiste, a parità di dati d'ingresso, **un solo passo successivo**; cioè, quando partendo dagli stessi dati in input, si ottengono i medesimi risultati in output;
3. **non ambiguo**, le operazioni devono poter essere interpretate nello stesso modo da tutti anche se l'esecutore è differente;
4. **generale**, cioè quando la soluzione è uguale per tutti i problemi della medesima classe.

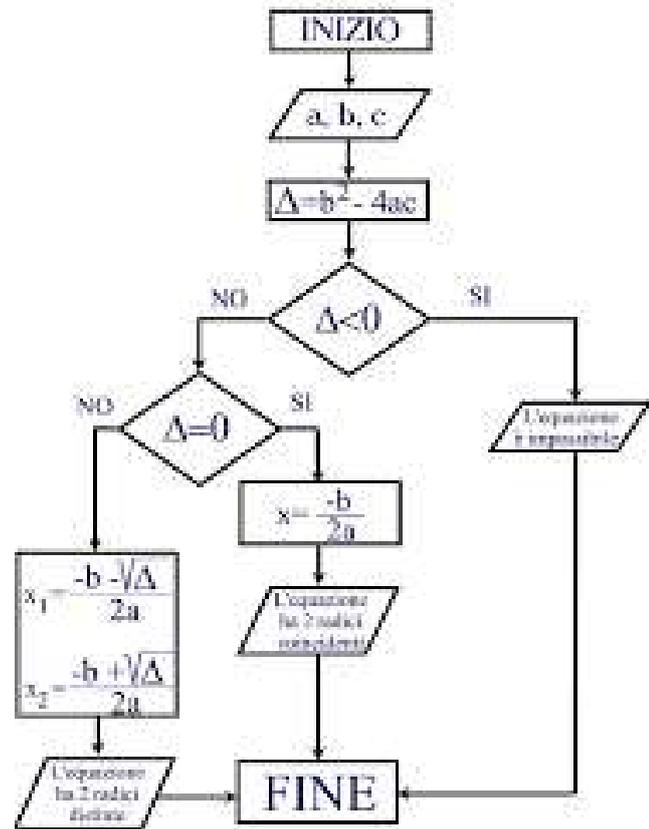
Inoltre

5. I passi che costituiscono lo schema devono essere **“elementari”**, ovvero non ulteriormente scomponibili (atomicità);
6. L'algoritmo deve essere **finito**, ossia composto da un numero definito di passi legati ad una quantità definita di dati in ingresso (finitezza);
7. L'esecuzione dello schema deve avvenire entro un tempo finito;
8. L'esecuzione dello schema algoritmico deve

condurre ad un unico risultato.

L'algoritmo quindi rappresenta l'insieme delle istruzioni che devono essere applicate per eseguire un'elaborazione o risolvere un problema e cioè qualsiasi procedimento che consenta, con un numero finito di passi eseguiti secondo un insieme finito di regole esplicite, di ottenere il risultato richiesto.

Esempio di algoritmo con diagramma di flusso
Soluzione di un'equazione di 2° grado
 $ax^2 + bx + c = 0$



Alcuni esempi di algoritmi

Algoritmo di ordinamento: Un algoritmo che ordina una lista di elementi in ordine crescente o decrescente.

Algoritmo di ricerca: Un algoritmo che trova un elemento specifico in una struttura di dati. Esempi includono la ricerca lineare e la ricerca binaria.

Ricerca lineare

La ricerca lineare è un algoritmo semplice utilizzato per trovare un elemento in una lista.

Funziona scorrendo la lista elemento per elemento, confrontando ciascun elemento con quello che si sta cercando.

Se trova una corrispondenza, restituisce la posizione dell'elemento; altrimenti, continua fino alla fine della lista.

Se l'elemento non è presente, restituisce un'indicazione che l'elemento non è stato trovato.

Ricerca binaria

La ricerca binaria è un algoritmo di ricerca efficiente utilizzato per trovare un elemento in una lista ordinata.

Funziona dividendo ripetutamente la lista a metà fino a trovare l'elemento desiderato o determinare che non è presente. Ecco come funziona:

Inizio: Si parte con l'intera lista ordinata.

Confronto: Si confronta l'elemento centrale della lista con l'elemento da cercare.

Divisione: Se l'elemento centrale è quello cercato, la ricerca termina. Altrimenti, se l'elemento cercato è minore dell'elemento centrale, si ripete la ricerca nella metà sinistra della lista. Se è maggiore, si ripete nella metà destra.

Ripetizione: Questo processo si ripete fino a trovare l'elemento o fino a quando la lista non può più essere divisa.

Algoritmo di compressione: Un algoritmo che riduce la dimensione di un file o di un insieme di dati.

Algoritmo di crittografia: Un algoritmo che protegge i dati trasformandoli in un formato illeggibile senza una chiave di decrittazione.