

ALGORITMO:

un algoritmo è una serie ordinate di azioni, queste non devono per forza provenire da un calcolatore.

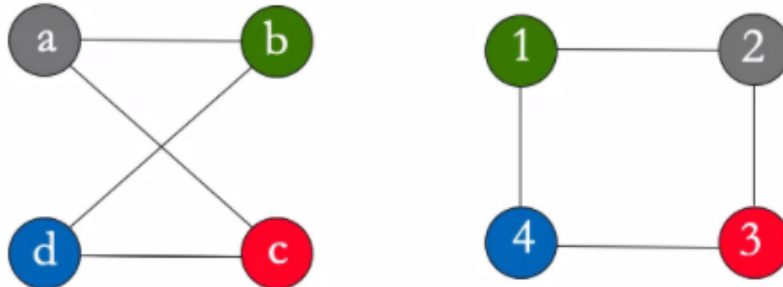
Più che una lista ordinata di elementi l'algoritmo è una lista numerata di passi che bisogna compiere. Nel linguaggio normale si danno per scontate moltissime cose che un calcolatore non può sapere e quindi è necessario che il programmatore pensi a tutte le possibili decisioni in seguito a un input.

determinare se due grafi sono isomorfi, cioè semplicemente se sono uguali.

Esempi (1)

Grafi

- Determinare se due grafi sono **isomorfi** (ovvero hanno le stesse caratteristiche in termini di nodi e archi)

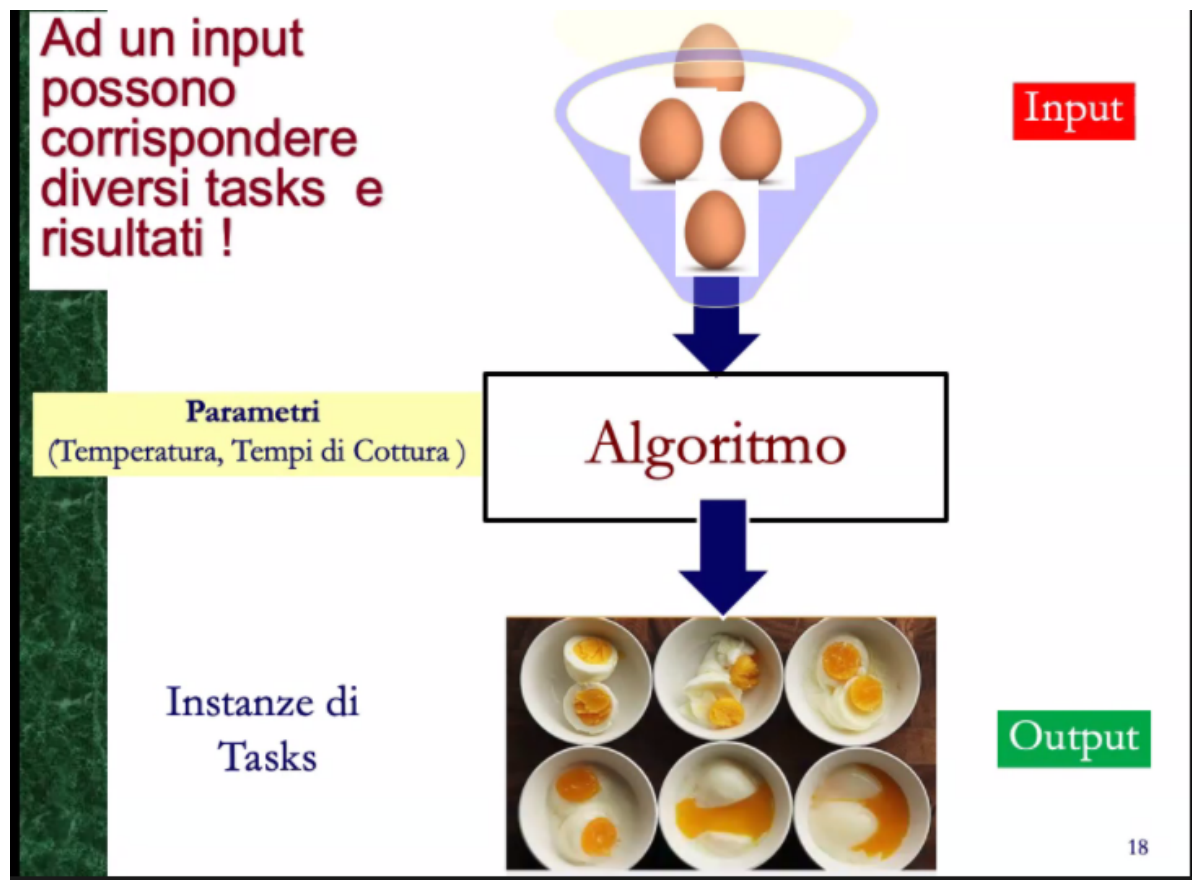


Potrebbero rappresentare il grafo degli utenti di facebook, in cui ogni ramo del grafo rappresenta un'amicizia. Quello che spendo dal nodo dipende da quello che sto cercando. In base a questa ricerca spendo quello che mi potrebbe servire di più.

I grafi si possono unire: facebook conosce i numeri di telefono delle persone da quando ha comprato whatsapp, e poi può fare ingerenza, cioè trovare e capire chi sono le persone attraverso l'analisi dei dati e dei grafi. Lo studio delle connessioni permette di capire il tuo numero.

Il grafo può avere tipologia simile, ma comunque uguale.

Se a un grafo si aggiungono dati e machine learning: questo permette un utilizzo dei dati e dell'oggetto del grafo molto potente.



Costruire un algoritmo che a partire dallo stesso input posso arrivare a uova diverse. Devo inserire il concetto di parametro. Concetto diverso se si usa il termine informatico o altre materie. I parametri condizionano l'esecuzione del task.

Come si descrive un algoritmo.

lista non ordinata di elementi, ma algoritmo è una sequenza precisa di passi. Manca un elemento: come selezionare, dove andare, come riuscire a fare....

Mancano tantissimi parametri, quindi c'è moltissima ambiguità nella specificità di un algoritmo.

Da questo parte tutto, perchè in base al linguaggio che uso avrò un certo vantaggio.

L'esecutore sarà la persona o un robot, che in più riesce a fare lui l'esecuzione, nel caso della persona invece sarà un'esecuzione assistita.

Descrizione per immagini: ottima informazione per supportare il risultato aspettato intermedio, ma non è sufficiente per tutta l'azione. Anche questo non è ottimale, perchè se no saremo tutti cuochi. Molto spesso si usano termini ambigui o si danno per scontati alcuni passi. Posso descrivere un microtask, o qualcosa di molto specifico.

Per descrivere l'algoritmo si può avere un linguaggio che sa descrivere da se tutte le azioni, es. manuale dei mobili ikea.

Altro passo: usare il linguaggio naturale per programmare. Non si usa un linguaggio in codice, ma usano il semplice linguaggio. Molte persone con questo linguaggio naturale adesso riescono a programmare.

esempi di algoritmi

algoritmo esula dal concetto di computer. Ma l'algoritmo deve essere stanziato, cioè affidato a un calcolatore.

Ci affideremo poi a algoritmi che necessitano di un computer. Ci affidiamo al fatto che per ogni algoritmo c'è la necessità di un calcolatore. Se state parlando con qualcuno dovete decidere con che lingua parlare. Tutto va bene, ma va condiviso.

Nella descrizione dell'algoritmo si deve arrivare a sequenze di istruzioni in linguaggio che formano un programma.

Spesso il lavoro del programmatore sarà capire il problema. Spesso il cliente avrà un'idea vaga e alcune volte pretenderà di sapere la soluzione. Il cliente ha una serie di requisiti.

I requisiti vanno riempiti, che poi allora se la risolve il cliente. Definire un linguaggio tra chi sta pagando e chi deve risolvere il problema. Con questo documento in cui spieghi come ti sei approcciato al problema e alla sua risoluzione diventerà un documento condiviso per avere una sorta di contratto.

Categorie di problemi

- Problemi di puro calcolo e conversione.
- problemi di decisione: decidere se una certa proprietà vale per un insieme di dati.
- problema di ricerca: trovare un elemento con le caratteristiche desiderate all'interno di un insieme.

Problema - Algoritmo - Task

per descrivere il problema useremo un linguaggio Lp (linguaggio naturale): molto rischioso, lascia molto comprensione. Quando il problema è complesso esistono linguaggi per questo.

Algoritmo: Pseudocodice, linguaggio tra il naturale e lo pseudo codice.

Programma: linguaggio Lt, linguaggio che descrive in linguaggio comprensibile al calcolatore.

Esempio: Gestione biblioteca

Contesto: biblioteca, libri sugli scaffali descritti dalle sue coordinate

Ogni libro si trova a una precisa variabile es. (posizione scaffale, posizione piano). La biblioteca è dotata di schedario.

scheda libro:

a questo punto possiamo formulare il problema in linguaggio naturale.

Formulazione del Problema

Data una biblioteca con uno schedario e con le caratteristiche descritte sopra si formuli un algoritmo per la ricerca di un libro:

INPUT

Seleziona il libro da richiedere

OUTPUT

Preleva il libro richiesto

36

definizione stereotipica: definizione del problema e del contesto. Definizione degli input e degli output.

Altri termini per la formulazione dell'algoritmo:

- o scrivo una sequenza di istruzioni specificate e comprensibili dall'istruttore, attraverso il cosiddetto linguaggio macchina, o in alto livello.
- quando si parla di processo *top-down* si parla di macro passi che poi si passano al raffinamento, quindi la creazione di micropassi. Si arriva fino alla comprensione del passo successivo per il funzionamento del programma.
- spesso il problema è talmente semplice che non c'è bisogno di raffinamenti successivi. Poi quando i passi diventano molto complicati c'è la necessità di formare questi macro-passi.

Formulazione dell'Algoritmo

posso farlo con il linguaggio naturale. Per mettere in ordine cosa viene prima e cosa dopo, inserisco un indice.

Formulazione dell' Algoritmo (1)

1. Acquisisci il libro da richiedere
2. Cerca la scheda del libro richiesto
3. Segnati numero scaffale e posizione
4. Cerca scaffale indicato
5. Accedi alla posizione indicata e preleva il libro
6. Scrivi i tuoi dati sulla scheda prestito

38

Nel primo passo prenderò in considerazione di tutti i passi per acquisire le informazioni del libro, nel secondo si troveranno tutti i passi per cercare la scheda del libro richiesto => passo più interessante.

nel Passo 2 ci sarà un altro algoritmo: Cerca la scheda del libro richiesto_

Esamina scheda e autore e se non corrisponde passa alla successiva e rifai. Se esaurisci le schede il libro cercato non esiste. **bisogna prevedere che l'algoritmo preveda tutti i casi**, che vanno gestiti.

Esiste una metodologia per fare queste verifiche. Spesso nella vita reale gli output possono essere infiniti, tecniche di campionamento.

Non è pensabile che ci sia differenza delle prestazioni dell'algoritmo in base all'input. Ci sarà differenza, ma bisogna ridurre al minimo.

Risolvere: Ricerca binaria, bicotomica => si prende e si divide in due nello schedario, e si cerca solo nella parte superiore o in quella inferiore se risulta prima o dopo.

Riesco in entrambi i casi a risolvere il problema, ma in un modo è più efficiente.

PROPRIETÀ DEGLI ALGORITMI:

- correttezza: l'algoritmo deve dare sempre la soluzione al compito, senza alcun errore negli spazi necessari, es. doppio risultato che si deve includere nel risultato. (primi programmi, questo è il punto)
- efficienza: algoritmo con meno risorse possibili. Un caso scansiona tutto lo schedario, nell'altro solo metà.

- Soluzione brute-force: soluzione lo è, ma usa moltissime risorse per un piccolo risultato
- riutilizzabile
- processo circolare: analizzare il problema, progettare una soluzione, esprimere l'algoritmo in linguaggio, mettere la macchina in grado di risolvere il problema, correggere eventuali errori.

linguaggio per descrivere un algoritmo o un programma:

si mantiene una sequenza, ma questo non vuol dire che le usiamo tutte e neanche che non ci possano essere salti.

COMPITO:

ESERCIZIO: Ordinamento mazzo di carte

“Sia data una serie completa di carte da gioco , 40 , da ordinare in modo che le picche precedano i fiori, e a loro volta precedano i quadri che a loro volta precedano i cuori (P-F-Q-C).”

Descriverlo in linguaggio naturale

Deconstruct, spaccare un grosso processo in grandi pezzi.

si usano costrutti linguistici: usato nei linguaggi di programmazione usando regole con il linguaggio.

A proposito dei costrutti, non esistono solo quelli sintattici, ma anche semantici.

Sintatti: insieme di regole che scrivono la generalizzazione delle frasi del linguaggio di programmazione.

Linguaggio di alto livello: utilizzo di variabili, intermedio.

Linguaggi Assembly: linguaggio assembler. Rende meglio se bisogna programmare ottimizzando su vari processori. sicuramente non è estensibile.

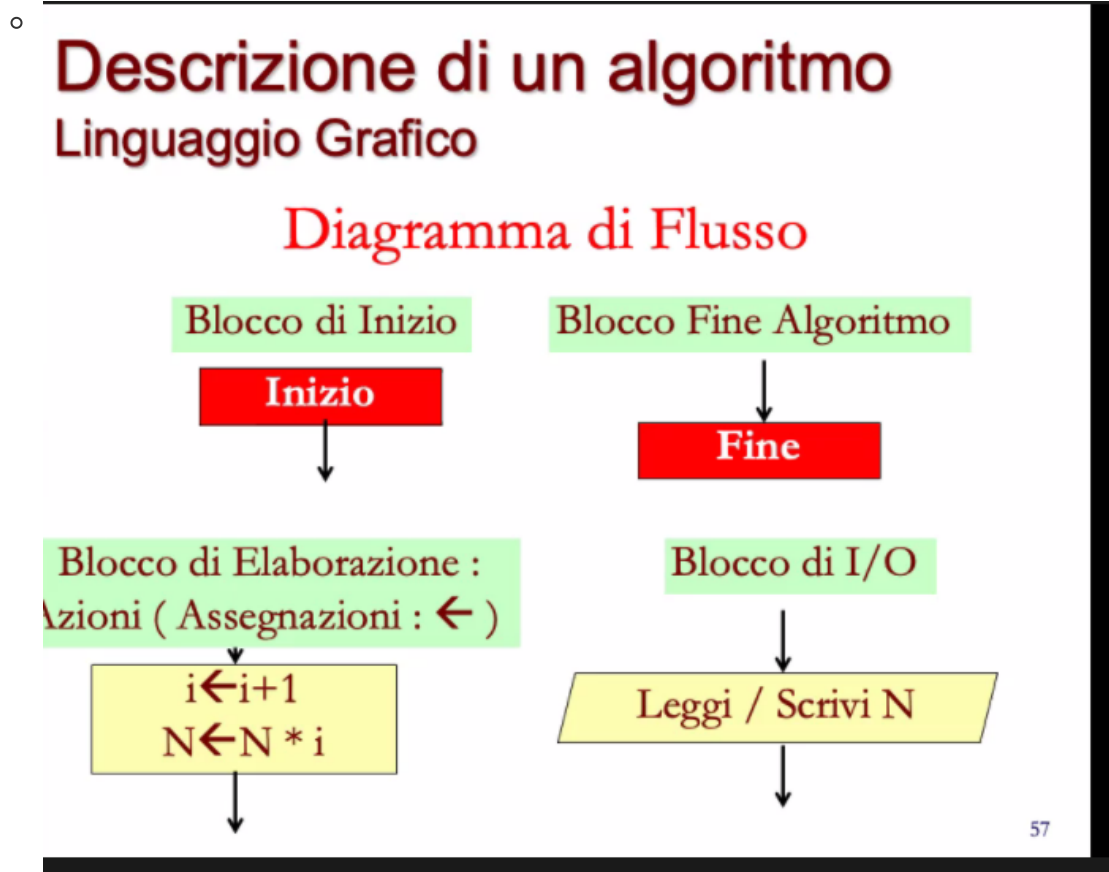
linguaggio macchina: 010101010101

Compilazione: esecuzione del programma per farlo diventare 0 e 1.

METODO GRAFICO PER CREARE ALGORITMI:

1. descrizione grafica di un algoritmo con diagramma di flusso:

- o hanno blocchi orientati, che hanno un senso.



- o inizio: unico nodo del grafo da cui si può solo uscire, ha una sola freccia uscente.
- o Fine: blocco con una sola freccia entrante. Rettangoli rossi per convenzione. Ci sono più modi di finire, ma di base rimane sempre uno entrante e zero uscente.
- o blocco istruzione: blocco con metacodice:
 - assegnazione: assegna il valore che c'è a destra alla variabile, cioè il valore che c'è a sinistra.
 - in questo caso sta incrementando i di uno
- o parallelogramma per input/output: leggendo o scrivendo/visualizzando
- o blocco di controllo: vari output (vero/falso) => più uscite
- o esempio sul quaderno
 - uso una variabile di servizio per calcolare i multipli di x, in modo che quando stampo 10 multipli, la variabile di servizio incrementi con me.
 - assegno a l = 1

2. conversione da base decimale a binario:

- msb: most significant bit; lsb: low significant bit
- da binare a decimale si usa il metodo dei resti. si usano i resti per costruire le cifre binarie. il numero meno significativo è l'ultimo resto, il più significativo l'ultimo. si parte dal meno significativo.

3. descriviamo l'algoritmo:

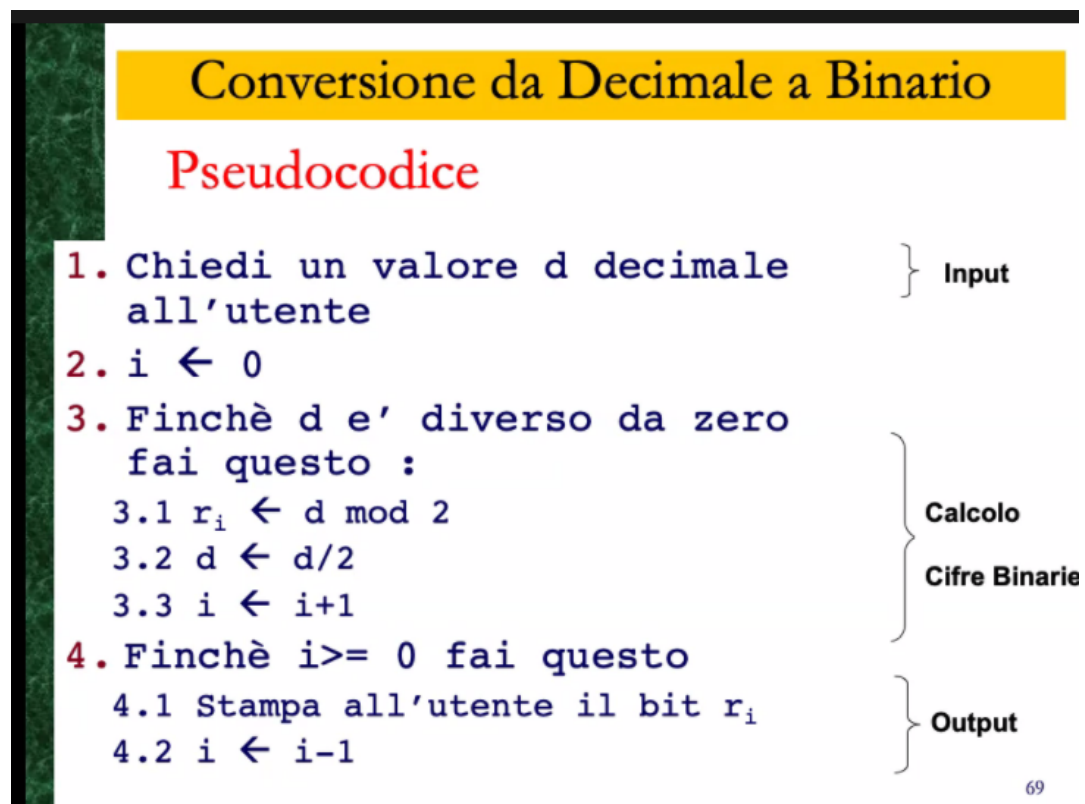
○

Descrizione dell' algoritmo

Linguaggio Naturale

Chiedi all'utente un numero d , poi dividilo per due ottenendo un quoziente ed un resto (il resto sarà zero oppure uno). Metti da parte il resto , che servirà dopo, e dividi di nuovo in due (cioè usa il quoziente della divisione precedente come dividendo della nuova divisione). Continua così finchè non arrivi ad avere un quoziente pari a zero. A questo punto metti insieme tutti i resti che hai ottenuto in ordine inverso, cioè dall'ultimo verso il primo.

- le cifre sono un numero binarie e devo ricordarmi di scriverle nell'ordine giusto.
- pseudocodice:



- passo 3 *finchè* => sto valutando un'istruzione, rispetto a prima che abbiamo usato un costrutto diverso.
- errore è che il 4.2 va prima del 4.1, se no un valore di i non avrà valore.
- trasformare lo pseudocodice in diagramma

anni '50 fanno i primi linguaggi di alto livello.

poi ci sono una serie di altri linguaggi per altri scopi.

Da una parte c'è un'evoluzione nei linguaggi di programmazione, che però contengono costrutti di base.

Nel corso di programmazione 2 si vedranno molti tipi di codici con cui poi possiamo fare quello che vogliamo.