ALGORITMI
E STRUTTURE DATI

Prof. Manuela Montangero

A.A. 2022/23

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

TERZA PARTE
Progettare algoritmi e pseudo-codice

Progettare di algoritmi

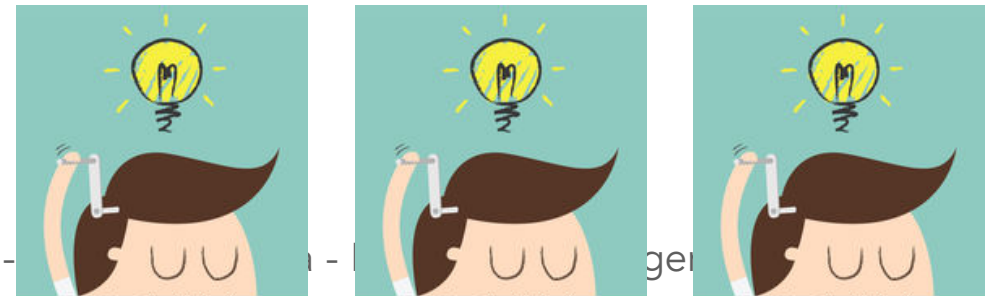
Cosa serve per progettare un ALGORITMO?

Dobbiamo sapere **quali** sono le **istruzioni** che l'esecutore preposto è in grado di comprendere ed eseguire

Dobbiamo scegliere un **formalismo** che ci permetta di descrivere la sequenza di istruzioni per l'elaboratore

- Linguaggio naturale
- Diagramma di flusso
- **Pseudo-codice**
- ...

IDEE



IDEE

Progettare di algoritmi

Dobbiamo sapere **quali** sono le **istruzioni** che l'esecutore preposto e' in grado di comprendere ed eseguire

Chi è il nostro esecutore?

Siamo interessati a fare eseguire gli algoritmi ad un computer

Perché?

Perché è più veloce degli umani

Perché commette meno errori degli umani
(assumendo che l'algoritmo sia corretto!)

Modello computazionale

MODELLO RAM (RANDOM-ACCESS-MEMORY)

astrazione dei computer moderni

- **Memoria principale infinita**
  - Ogni cella di memoria può contenere un *dato* o una *istruzione* (quantità finita)
  - Il tempo di accesso ad una cella di memoria è lo stesso per ogni cella
- **Singolo processore**
  - In una unità di tempo può eseguire una operazione tra: *lettura*, *esecuzione* di un'*istruzione*, *scrittura*
  - Istruzioni: operazioni logico-aritmetiche, assegnamento, accesso a puntatore

Formalismo di descrizione degli algoritmi

- **Diagrammi di flusso**

  (ne avete parlato a Programmazione I?)
- **Pseudo-codice**
  - Non è un linguaggio direttamente eseguibile da un computer
  - È vicino a molti linguaggi di alto livello
  - Ammette mezzi espressivi diversi

    (es, frase in italiano "scegli un numero nell'insieme S")
  - Non si occupa di problemi di basso livello
  - **Permette di concentrasi sulla soluzione ad alto livello**

Pseudo-codice

ATTENZIONE

In questo corso, le soluzioni dei problemi verranno formalizzate utilizzando pseudo-codice
ed è quindi FONDAMENTALE padroneggiare alcuni concetti basilari di PROGRAMMAZIONE

- Costrutti principali di un linguaggio di programmazione
- Array e matrici
- Variabili locali, variabili globali e loro differenze
- La visibilità (o scope) di una variabile
- Sottoprogrammi: funzioni e procedure e loro differenze
- Passaggio di parametri per valore e per riferimento

Potete fare riferimento alla pagina del corso di Programmazione I

Pseudo-codice

Dato un **problema** specificato da una coppia **(INPUT,OUTPUT)**
formalizzeremo la soluzione al problema scrivendo lo
pseudo-codice di una

FUNZIONE/PROCEDURA PRINCIPALE che
prende come parametri gli INPUT del problema
e
restituisce l'OUTPUT del problema
o manipola opportunamente l'input

Analogo di `main` in C, ma senza la necessità di generare l'input

La FUNZIONE/PROCEDURA che risolve il problema può chiamare
dei **sottoprogrammi** (funzioni o procedure)
ed
è scritta usando la **sintassi** prevista dal nostro pseudo-codice

Pseudo-codice

Istruzioni principali

Assegnamento: `:=`

Costrutti condizionali: `if-then`, `if-then-else`

Costrutti iterativi: `while..do`, `repeat..until`, `for..to..[step..]`, `for..downto..[step..]`

+ indentazione

Terminazione: `return`, `return(lista valori)`

Errore: `error`

Commento: `//`

Definiremo altri comandi o costrutti quando ne avremo bisogno

Operatori logico/artimentici: `/+,-,*,/`… `<,>,=` … **`AND,OR,NOT`** …

Pseudo-codice

ATTENZIONE

nel corso non faremo uso di **BREAK** e **CONTINUE**

e

l'uso di questi due comandi nelle soluzioni del compito

E' VIETATO

(la valutazione dell'esercizio sarà **ZERO**)

FINE TERZA PARTE

ALGORITMI
E STRUTTURE DATI

Prof. Manuela Montangero

A.A. 2022/23

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

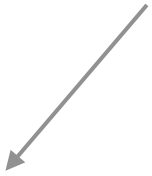
QUARTA PARTE
Strutture Dati

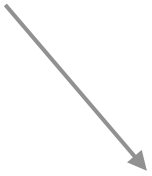
STRUTTURE DATI

Gli algoritmi elaborano i dati di input per produrre l'output,
e spesso hanno anche bisogno di produrre ed elaborare una serie di dati intermedi

In molti casi organizzare questi dati dandogli una struttura permette di progettare algoritmi
più semplici, più eleganti, più efficienti....

STRUTTURA DATI
"Insieme di dati indirizzabile con un solo nome organizzati in un modo specifico"





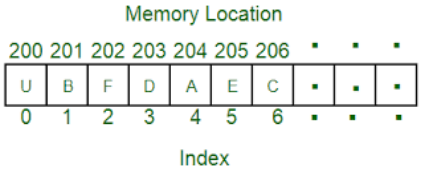
ELEMENTARE

ASTRATTA

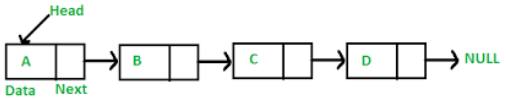
STRUTTURE DATI

ELEMENTARE

Struttura dati le cui caratteristiche dipendono in modo stretto dal modo in cui i dati vengono effettivamente memorizzati



Array, Liste, Alberi, Grafi...



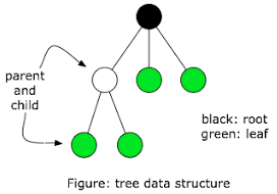
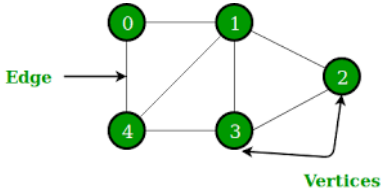


Figure: tree data structure



ASTRATTA

Struttura dati che offre una visione ad alto livello dei dati e delle operazioni ammissibili su di essi, senza fornire dettagli su come siano rappresentati i dati e come siano implementate le operazioni su di essi

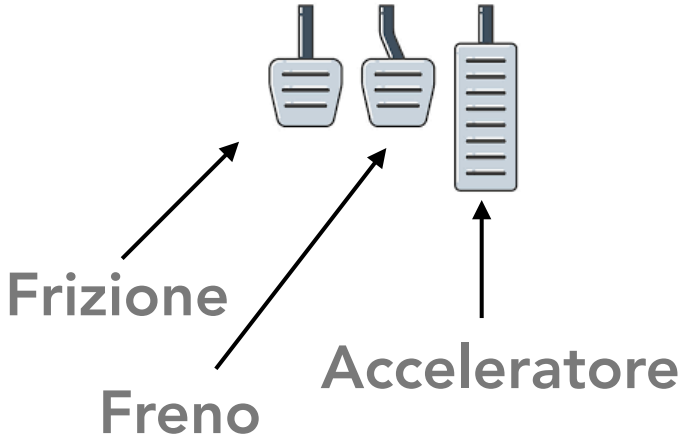
STRUTTURE DATI ASTRATTE

Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

ANALOGIA di astrazione

Pedaliera di un'automobile



STRUTTURE DATI ASTRATTE

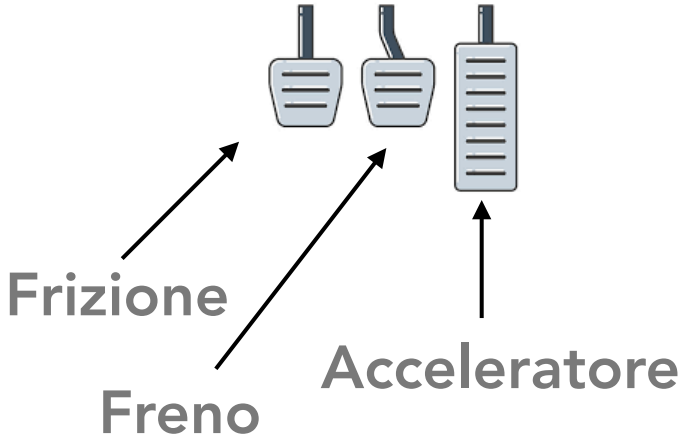
Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

ANALOGIA di astrazione

Pedaliera di un'automobile

Questo ci basta per poter usare la pedaliera



Proprietà: 3 pedali: la Frizione sta a sinistra, il Freno sta al centro, l'Acceleratore sta a destra, freno e acceleratore non possono essere premuti contemporaneamente.

Operazioni:

- Premendo l'acceleratore l'auto va più veloce
- Premendo il freno l'auto rallenta
- Premendo la frizione e' possibile cambiare marcia

STRUTTURE DATI ASTRATTE

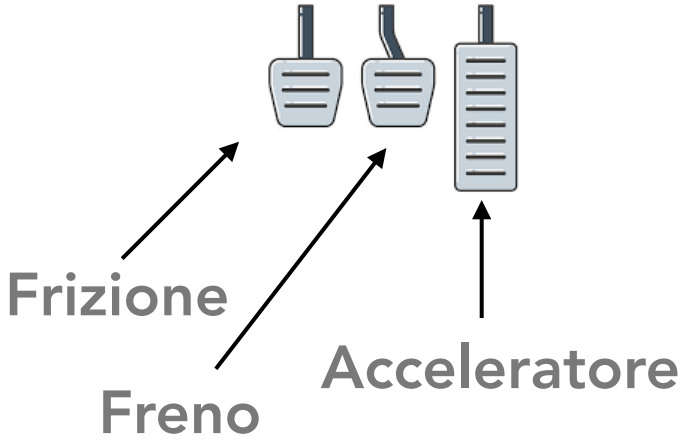
Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

ANALOGIA di astrazione
Pedaliera di un'automobile

Realizzazioni diverse portano a prestazioni diverse…

Ma come fa a succedere quello che deve accadere quando premo un pedale?



Dipende da come e' realizzata la pedaliera…

STRUTTURE DATI ASTRATTE

Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

ESEMPIO

Struttura dati astratta **PILA** (di cui parleremo di nuovo)

- Memorizza un insieme di dati generalmente dello stesso tipo
- Ammette le seguenti operazioni:
  - ***stack_init()***: inizializza una pila vuota
  - ***empty()***: restituisce vero se e solo se la pila e' vuota
  - ***push(val)***: inserisce il valore *val* nella pila
  - ***pop()***: restituisce l'ultimo valore inserito nella pila e lo elimina dalla pila
  - ***top()***: restituisce l'ultimo valore inserito nella pila, ma non lo elimina dalla pila

STRUTTURE DATI ASTRATTE

Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

Una struttura dati astratta può essere
utilizzata
servendosi delle operazioni definite dalla struttura
senza preoccuparsi di come queste siano
implementate

Utilizzare strutture dati astratte appropriate
può permettere di migliorare le prestazioni
di algoritmi per la soluzione di problemi

STRUTTURE DATI ASTRATTE

Una **struttura dati astratta** di definisce indicando:

- Le **proprietà** che deve soddisfare l'insieme di dati memorizzato nella struttura
- Le **operazioni** (***primitive***) ammesse sull'insieme di dati

Una struttura dati astratta può essere
implementata
in tanti modi diversi

Diverse implementazioni possono avere prestazioni diverse
(alcune operazioni possono essere più efficienti o inefficienti
a seconda dell'implementazione)

Pseudo-codice: strutture dati astratte

Implementazione strutture dati

Lezione e "Cormen et al." —> approccio "programmazione imperativo" (come in C)

Primitive come funzioni/procedure che prendono come parametri variabili di tipo la struttura dati e altri valori necessari

Bertossi, Montresor —> approccio "programmazione ad oggetti" (come in C++)

Primitive come metodi di un oggetto che implementa la struttura dati e prendono come parametri valori necessari, ma non la struttura dati

Esempio:
- `Q` oggetto queue (coda)
- `pop( )` metodo dell'oggetto queue

Invocazione metodo: `Q.pop( )`

FINE QUARTA PARTE

ALGORITMI
E STRUTTURE DATI

Prof. Manuela Montangero

A.A. 2022/23

UNIMORE
UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

QUARTA PARTE
Cosa faremo nel corso e consigli per l'esame

Algoritmi e strutture dati

Cosa faremo durante il resto del corso?

Studieremo algoritmi noti per problemi fondamentali

- Processo risolutivo
- Studio correttezza
- Studio efficienza

Studieremo implementazioni standard di strutture dati astratte fondamentali

- Definizione
- Implementazioni delle operazioni
- Studio efficienza e correttezza

Studieremo tecniche standard utilizzate nella progettazione di algoritmi

Algoritmi e strutture dati

Cosa vi verrà chiesto nella **prima parte** della prova scritta?

- Risolvere problemi che abbiamo visto a **lezione utilizzando gli algoritmi visti a lezione** (nelle implementazioni viste a lezione)

Come ci si prepara per la prima parte della prova scritta?

- Si **studiano** gli algoritmi visti a lezione (ovvero, si capisce perché risolvono il problema e perché sono state fatte determinate scelte)
- Si eseguono gli algoritmi (esattamente) su istanze del problema
- Verranno forniti degli esercizi con soluzione

Algoritmi e strutture dati

Cosa vi verrà chiesto nella **seconda parte** della prova scritta?

- Risolvere problemi di difficoltà analoga a quelli visti a lezione, ma che non abbiamo mai studiato (insieme)

Come ci si prepara per la seconda parte della prova scritta?

Algoritmi e strutture dati

Cosa vi verrà chiesto nella **seconda parte** della prova scritta?

- Si studiano gli algoritmi visti a lezione per capire la logica che c'è dietro e perché sono corretti ed efficienti
- Si fanno gli esercizi proposti su moodle
- Si fanno le prove d'esame vecchie pubblicate su moodle
- Si possono fare esercizi extra presenti sui libri o sul Web
- Si possono provare ad implementare le soluzioni trovate agli esercizi proposti
- Si possono discutere con i colleghi le soluzioni trovate e quelle pubblicate
- Si possono aiutare i colleghi in difficoltà a svolgere gli esercizi
  (spiegare ad altri le proprie idee serve per chiarirsi le idee e capire se si e' in grado di formulare spiegazioni coerenti e comprensibili)

Algoritmi e strutture dati

Come si affrontano gli esercizi proposti su moodle?

1. Leggere attentamente il testo per capire il problema
2. Fare qualche esempio per capire il problema
3. Pensare ad una soluzione, mettendo a fuoco i passaggi principali
4. Scrivere lo pseudo-codice relativo alla soluzione proposta
5. Provare ad eseguire lo pseudo-codice su qualche istanza del problema per vedere se, almeno su quelle, funziona
6. Provare a pensare ad istanze che sono casi limite, per vedere se la soluzione proposta funziona
7. Provare a dare una spiegazione generale del perché la soluzione proposta funziona sempre (una dimostrazione sarebbe perfetta, ma bastano delle argomentazioni convincenti)
8. SOLO A QUESTO PUNTO: guardare la soluzione
9. Se la soluzione proposta è come la soluzione (a meno di di dettagli implementativi) allora questo esercizio è risolto
10. Altrimenti….

Algoritmi e strutture dati

Come si affrontano gli esercizi proposti su moodle?

10. Altrimenti….
11. Rileggere con attenzione la soluzione
12. Capire (eventualmente con l'aiuto della soluzione) perché la soluzione proposta non funziona
13. SENZA ausilio della soluzione, proporre una nuova soluzione al problema (a questo punto l'idea dovrebbe essere corretta, se si è capita la soluzione, ma si deve essere in grado di scrivere lo pseudo-codice corretto senza "sbirciare" la soluzione)
14. Confrontare la nuova soluzione proposta con la soluzione
15. Ripetere partendo dal punto 5

Quando vi chiedo se avete fatto gli esercizi su moodle
e mi rispondete di si,
mi aspetto che abbiate fatto questo tipo di lavoro
e che ne siate usciti come al punto 9

Algoritmi e strutture dati

Cosa vi verrà chiesto nella **seconda parte** della prova scritta?

- Rispondere a delle domande teoriche sugli argomenti visti a lezione

Come ci si prepara per la seconda parte della prova scritta?

Si studia quello che è stato fatto a lezione su appunti, libro e altro materiale fornito,
senza studiare a memoria, ma capendo il perché di quello che si sta studiando, tanto da essere in grado di spiegarlo nuovamente a qualcun'altro

FINE QUINTA PARTE