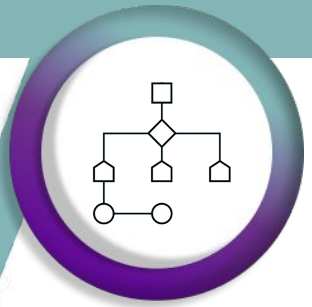
ALGORITMO

DEAF YOUNG CODE

Numero di progetto: 2023-2-IT03-KA220-YOU-000179130

Co-funded by
the European Union

Piano di Lezione Avanzato



- Definire un algoritmo in senso tecnico e spiegarlo in modo semplice.
- Comprendere l'efficienza degli algoritmi: sapere cosa significano "complessità temporale" e "complessità spaziale".
- Identificare i tipi comuni di algoritmi: ordinamento, ricerca, ricorsivi, iterativi, ingordi (greedy), ecc.
- Analizzare un algoritmo passo dopo passo per verificarne la correttezza e l'efficienza.
- Scrivere semplici algoritmi in pseudocodice o diagrammi di flusso.

Co-funded by
the European Union

Video

DEAF YOUNG CODE

Co-funded by
the European Union

Cos'è un Algoritmo?

DEAF YOUNG CODE

Definizione Tecnica::

Un algoritmo è una sequenza finita di istruzioni ben definite per risolvere un problema.

Co-funded by the European Union

Caratteristiche Chiave

DEAF YOUNG CODE

1. Input: I dati di partenza.
2. Output: Il risultato o la soluzione.
3. Finitezza: Deve terminare dopo un numero limitato di passaggi
4. Determinatezza: Ogni passaggio è chiaramente definito.
5. Efficacia: Ogni passaggio può essere eseguito in una quantità di tempo ragionevole.

Basi degli algoritmi

DEAF YOUNG CODE

Come funzionano gli algoritmi

- **Definire chiaramente il problema** → Cosa vogliamo risolvere?
- **Progettare una procedura passo-passo** → Quali sono le istruzioni per raggiungere una soluzione?
- **Testare con esempi** → Controllare se l'algoritmo funziona con input di esempio.
- **Analizzare l'efficienza** → Quanto è veloce? Quanta memoria utilizza?
- **Perfezionare e ottimizzare** → Può essere più veloce o usare meno memoria?

Diagramma dell'Algoritmo (Diagramma di flusso)

DEAF YOUNG CODE

Un diagramma di flusso è un modo visivo per mostrare un algoritmo passo dopo passo. Aiuta a vedere la logica, non solo a leggerla.

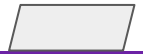
Elemento	Simbolo	Funzione	Esempio
Inizio / Fine	Ovale 	Mostra dove inizia o finisce l'algoritmo	"Inizio" / "Fine"
Processo / Passaggio	Rettangolo 	Un'azione o istruzione	"Somma due numeri"
Decisione / Condizione	Rombo 	Una domanda che cambia il flusso (Sì/No)	"a > b?"
Input / Output	Parallelogramma 	Dati che entrano o escono	"Inserisci numeri" / "Mostra risultato"
		Mostra la	

Diagramma dell'Algoritmo

Come leggere un diagramma di flusso

- Inizio → Trova l'ovale "Inizio".
- Input → Guarda quali informazioni entrano nel sistema (es. numeri, testo).
- Processo → Segui ogni rettangolo. Queste sono azioni o passaggi.
- Decisione → Quando raggiungi un rombo, pone una domanda a risposta
  Sì/No. Se "Sì", segui una freccia.
  Se "No", segui l'altra.
- Ciclo → Quando una freccia torna a un passaggio precedente, significa "ripeti".
- Output → Alla fine, vedi il risultato o la risposta.
- Fine → Il processo si ferma.

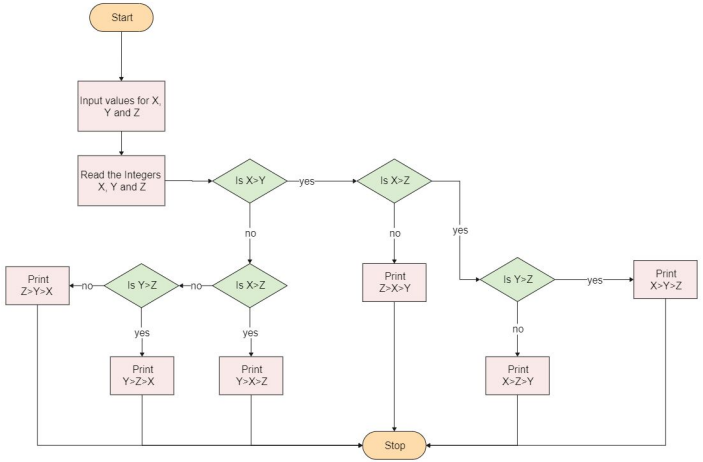
Esempi: Ordinamento di Numeri

Problema: Ordinare i numeri [4, 2, 7] dal più piccolo al più grande.

1. Diagramma di flusso

(Visualizzazione passo-passo)

```
Inizio
  |
  v
Inserisci numeri: 4, 2, 7
  |
  v
Confronta i primi due numeri:
 4 è > 2? Si → Scambia
  |
  v
Numeri attuali: 2, 4, 7
  |
  v
Confronta i numeri successivi: 4 and 7
4 è > 7? No → Mantieni l'ordine
  |
  v
Output numeri ordinati: 2, 4, 7
  |
  v
Fine
```



Suggerimento per studenti sordi:

- **Usa frecce e riquadri per ogni passaggio.**
- **Evidenzia l'azione di scambio con un colore diverso.**
- **Mostra chiaramente input → processo → output.**

Co-funded by the European Union

Esempi: Ordinamento di Numeri

Problema: Ordinare i numeri [4, 2, 7] dal più piccolo al più grande.

2. Pseudocodice (Versione di testo semplice)

```
Algoritmo Ordina Tre Numeri:
  Input: a, b, c
  Se a > b allora
      Scambia a e b
  Se b > c allora
      Swap b and c
  Se a > b allora
      Scambia a e b
  Output: a, b, c
Fine Algoritmo
```

Note:

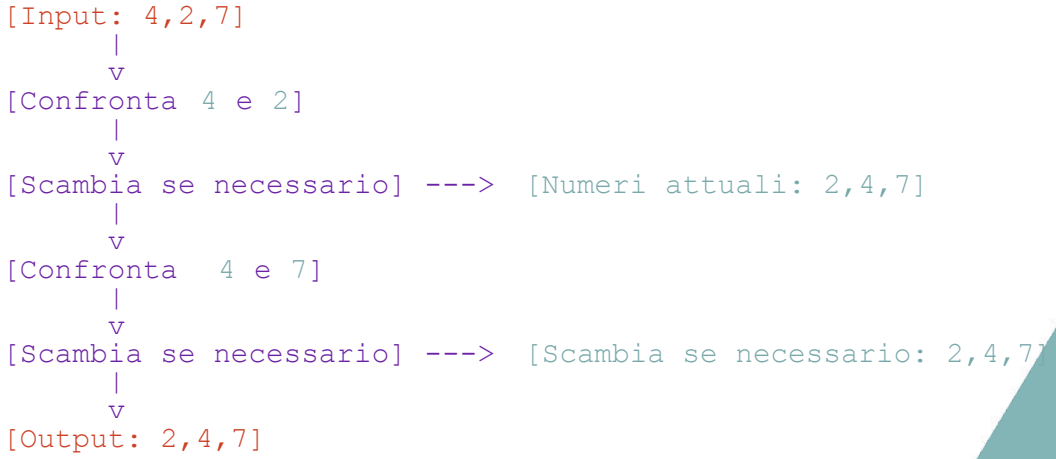
- **Lo pseudocodice è composto da istruzioni come l'inglese, non da un linguaggio di programmazione.**
- **La chiarezza passo-passo è fondamentale.**

Co-funded by the European Union

Esempi: Ordinamento di Numeri

Problema: Ordinare i numeri [4, 2, 7] dal più piccolo al più grande.

3. Diagramma (Panoramica visiva)



Co-funded by
the European Union

Punti Chiave

- Diagramma di flusso = mappa dettagliata e passo-passo dell'algoritmo.
  - Include decisioni, cicli, input, output.
  - Ottimo per scrivere pseudocodice o programmare in seguito.
- Diagramma = panoramica visiva generale, sintesi semplice.
  - Mostra le azioni principali e il flusso senza tutti i passaggi.
  - Aiuta a comprendere rapidamente la logica, specialmente per chi apprende visivamente

Co-funded by
the European Union

Teoria degli algoritmi: concetti importanti

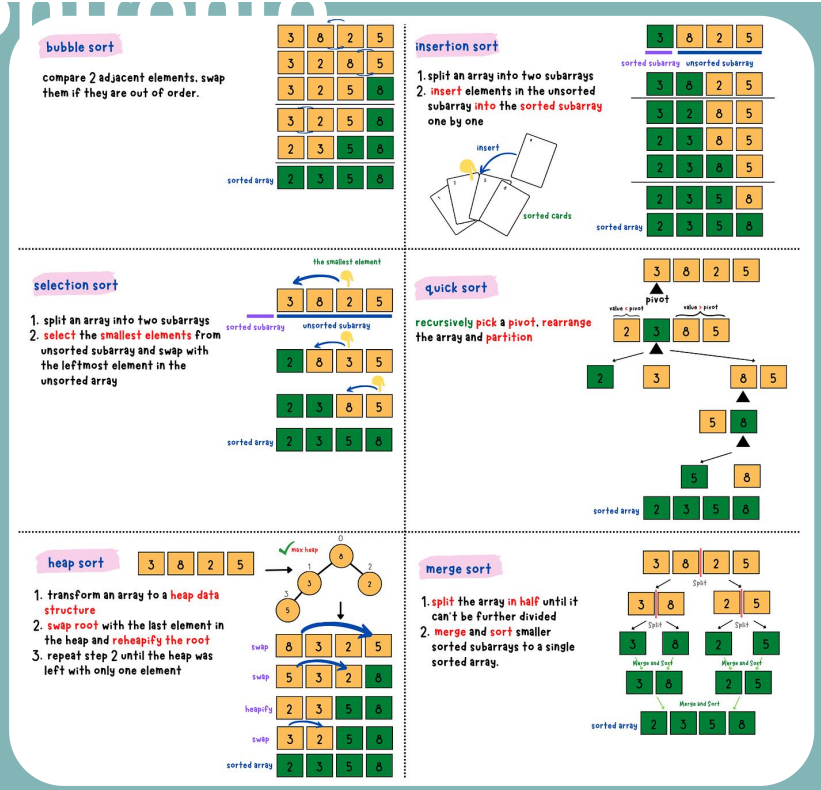
- **Complessità temporale (Notazione Big O): Misura quanto velocemente viene eseguito un algoritmo all'aumentare delle dimensioni dell'input.**
- **Complessità spaziale: Misura quanta memoria utilizza un algoritmo.**
- **Tipi comuni di algoritmi:**
  - Ordinamento: Bubble sort, merge sort, quicksort.
  - Ricerca: Ricerca lineare, ricerca binaria.
  - Ricorsione: L'algoritmo richiama se stesso (es. fattoriale).
  - Algoritmi Ingordi (Greedy): Fanno la scelta migliore in ogni passaggio (es. percorso più breve).
  - Programmazione dinamica: Suddivide i problemi in sottoproblemi più piccoli.
- **Strutture dati: Gli algoritmi spesso necessitano di liste, array, alberi o grafi.**

DEAF YOUNG CODE

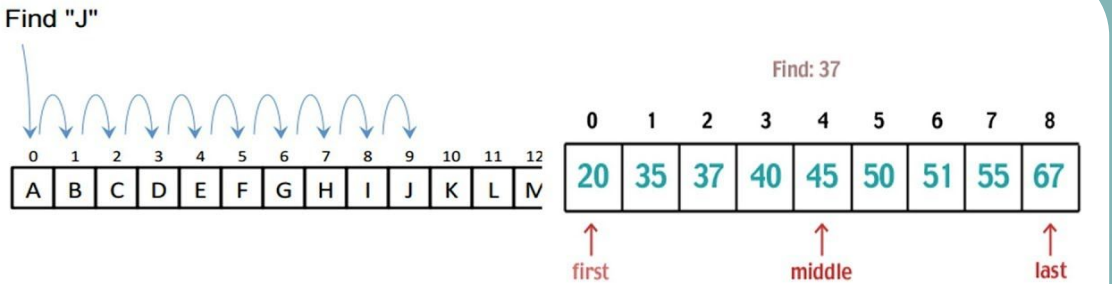
Co-funded by the European Union

Ordinamento basato sul confronto

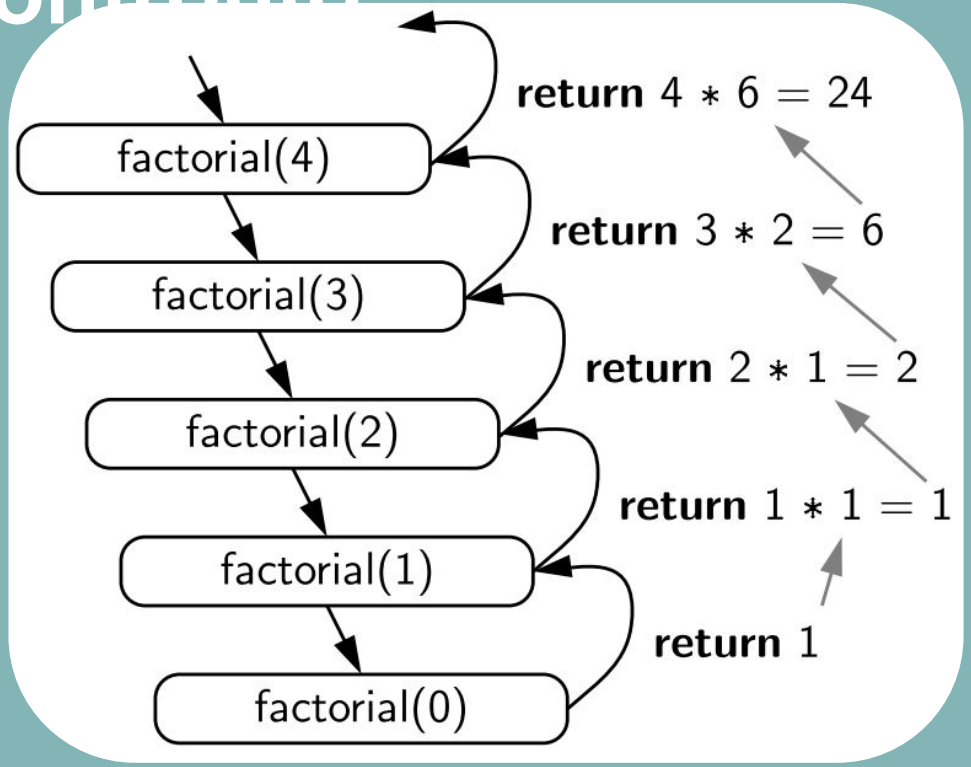
DEAF YOUNG CODE



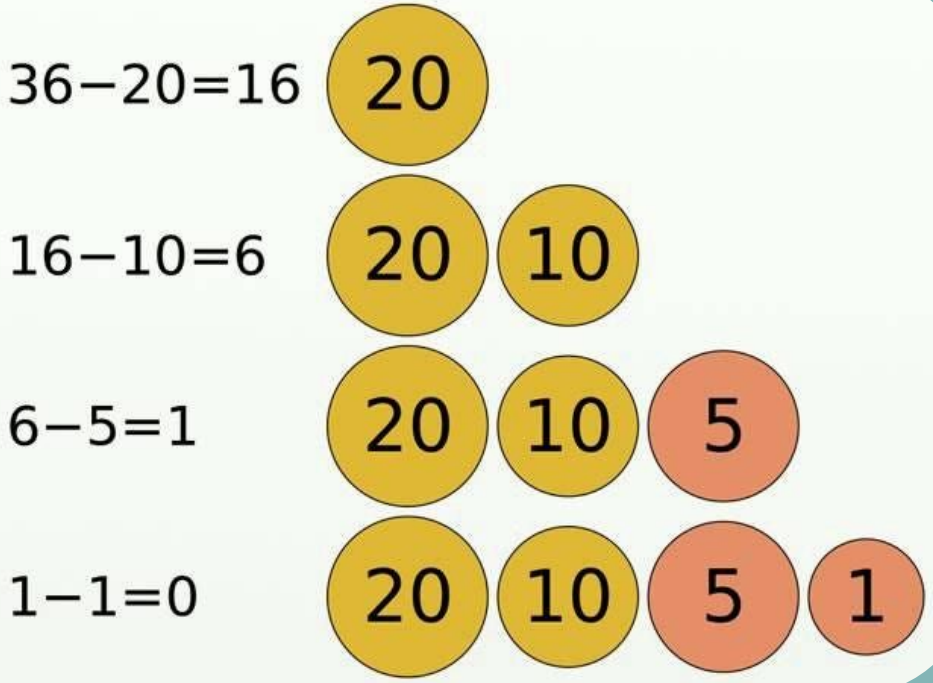
Ordinamento basato sul confront



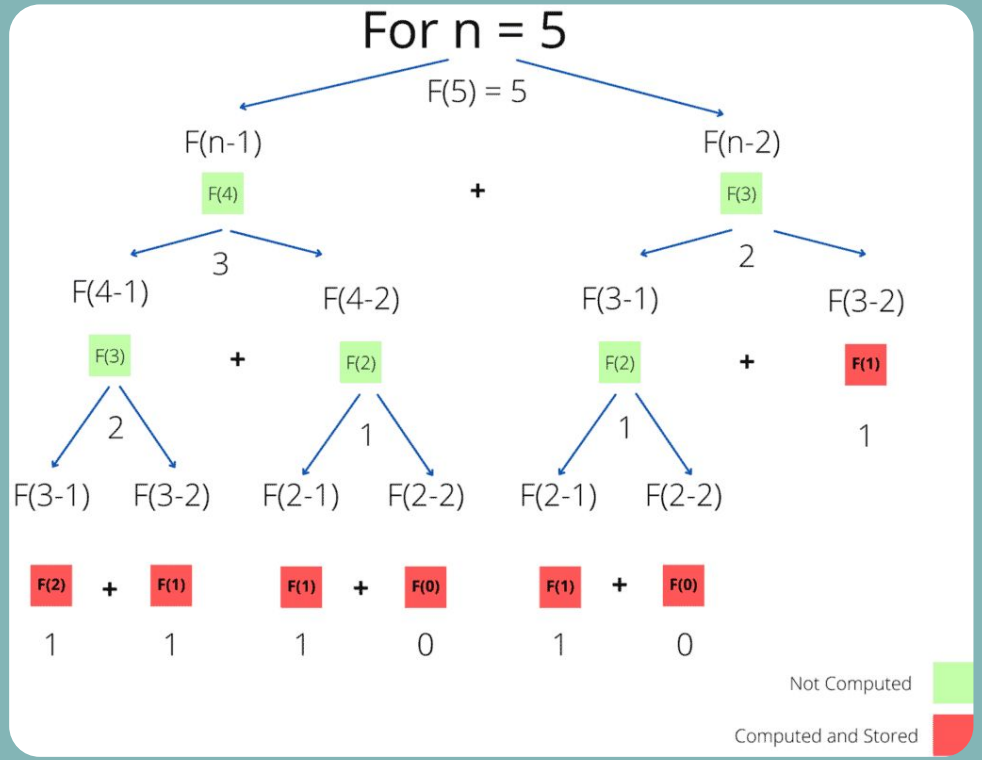
Ordinamento basato sul confronto



Ordinamento basato sul confronto



Ordinamento basato sul



Idea per Esercizio Interattivo (Apprendimento Pratico)

Obiettivo:

- Aiutare gli studenti a fare pratica nella progettazione di algoritmi in modo divertente e visivo.
- Rafforzare la comprensione di passaggi, decisioni, cicli ed efficienza.

Idea per Esercizio Interattivo (Apprendimento Pratico)

DEAF YOUNG CODE

Passaggio 1: Scegli un Compito Semplice

- Preparare il tè
- Ordinare i libri per altezza
- Routine mattutina (svegliarsi → lavarsi i denti → fare colazione → vestirsi)
- Preparare lo zaino
- Ordinare la frutta per dimensione

Idea per Esercizio Interattivo (Apprendimento Pratico)

Passaggio 2: Disegna un Diagramma di Flusso

- Istruzioni per gli studenti:
- Inizio → Disegna il simbolo di inizio (ovale).
- Input / Output → Mostra i dati o il risultato (parallelogramma).
- Azioni / Passaggi → Usa i rettangoli per ogni azione.
- Punti di decisione → Usa i rombi per le domande Sì/No.
- Cicli → Disegna frecce che tornano indietro per ripetere i passaggi se necessario.
- Fine → Mostra il risultato finale o il completamento (ovale).

Idea per Esercizio Interattivo (Apprendimento Pratico)

Passaggio 3: Condividi e Confronta

- Chiedi a ogni studente o gruppo di presentare il proprio diagramma di flusso.

Discutere:

- Hanno incluso tutti i passaggi importanti?
- Ci sono passaggi non necessari che possono essere rimossi?
- Il compito può essere svolto più velocemente o in modo più efficiente?

Idea per Esercizio Interattivo (Apprendimento Pratico)

DEAF YOUNG CODE

Passaggio 4: Sfida Opzionale (Efficienza)

- Introduci l'idea dell'efficienza dell'algoritmo:
  - Meno passaggi = algoritmo più veloce
  - Ripetere passaggi non necessari = algoritmo più lento

Idea per Esercizio Interattivo (Apprendimento Pratico)

Passaggio 4: Sfida Opzionale (Efficienza)

- Esempio: Ordinare 3 libri
  - **Studente A**: Controlla tutte le coppie due volte → 6 passaggi
  - **Studente B**: Solo scambi intelligenti → 3 passaggi
- Chiedi**: Quale algoritmo è migliore? Perché?**
- Mostra che **anche semplici compiti quotidiani possono avere algorit**mi "più efficienti".

Idea per Esercizio Interattivo (Apprendimento Pratico)

DEAF YOUNG CODE

Passaggio 5: Bonus Pratico (Apprendimento Cinestetico)

- Usa **oggetti reali**: libri, tazze, post-it.
- Gli studenti mettono in **atto i passaggi fisicamente**, seguendo il proprio diagramma di flusso.
- Aiuta chi apprende in **modo visivo e tattile a comprendere la sequenz**a, i cicli e le decisioni.

Grazie per l'attenzione!

DEAF YOUNG CODE

Restate sintonizzati per la prossima presentazione!

Finanziato dall'Unione Europea. Le opinioni espresse appartengono tuttavia al solo o ai soli autori e non riflettono necessariamente quelle dell'Unione Europea o dell'Agenzia Esecutiva Europea per l'Istruzione e la Cultura (EACEA).Né l'Unione Europea né l'EACEA possono esserne ritenute responsabili.

Numero di progetto: 2023-2-IT03-KA220-YOU-000179130

Co-funded by the European Union