

Cosa studieremo

- **Algoritmi** = descrizione precisa di una sequenza di azioni che devono essere eseguite per giungere alla risoluzione di un problema
 - Analisi / progetto / codifica
 - Analisi dell'efficienza
- **Strutture dati** = è fondamentale che i dati siano ben organizzati e strutturati in modo che l'algoritmo li possa elaborare efficientemente



1

Strutture dati: Tipo di dato

- I linguaggi di programmazione di alto livello consentono di far riferimento a posizioni nella memoria principale tramite nomi descrittivi (**variabili**) anziché indirizzi numerici
- **Dato**: un particolare valore che una variabile può assumere
- **Tipo di dato**: modello matematico che caratterizza l'insieme di valori che una variabile può assumere, e le operazioni che possono essere eseguite su di essa

2

Strutture dati: Tipi di dato primitivi

- Esempi:
 - **Intero** (integer):
 - ♦ dati numerici costituiti da numeri interi
 - ♦ Operazioni: aritmetiche e confronto
 - **Reali** (float, real)
 - ♦ dati numerici costituiti da numeri non interi
 - ♦ Operazioni simili a interi
 - **Booleani** (boolean)
 - ♦ Vero/falso
 - ♦ Operatori dell'algebra booleana, confronto
 - **Caratteri** (char)
 - ♦ Dati alfanumerici
 - ♦ Operazioni: concatenazione, confronto

3

Strutture dati: *Composiz. di dati semplici*

- Un modo per memorizzare e organizzare i dati e rendere efficiente l'accesso e la modifica dei dati stessi
- Esempi: array, matrici, grafi, alberi, tabelle hash, heap, liste, code, pile, ...
- Una struttura dati consiste di:
 1. un modo sistematico di organizzare i dati
 2. un insieme di operatori che permettono di manipolare gli elementi della struttura

4

Tipo di Dato Astratto (ADT)

- Un tipo di dato astratto (ADT) è un modello astratto della struttura dati che specifica il tipo dei dati da cui è costituita, le operazioni che possono essere fatte su quella struttura dati e i parametri richiesti per effettuare tali operazioni.
- Un ADT specifica che cosa fa un'operazione ma non il come.
- L'insieme delle operazioni offerte dall'ADT rappresentano la sua interfaccia pubblica (**public interface**).

5

Tipi di strutture dati

- **Statiche:** la dimensione è definita al momento della creazione. Una volta terminato lo spazio allocato, è necessario creare un'altra struttura di dimensione maggiore dello stesso tipo e copiarvi dentro il contenuto della prima
 - Array, record
- **Dinamiche:** la dimensione della struttura dati può variare nel tempo senza limite. L'unico limite è la quantità di memoria disponibile nella macchina su cui andrà in esecuzione l'applicazione
 - Liste, code, pile, alberi, grafi

Possibile spreco

6

Tipi di strutture dati

- **Lineari:** i dati sono disposti in sequenza e possono essere nominati come primo, secondo, terzo, ...
- **Non lineari:** i dati non sono disposti in sequenza *Alberi
Grafici*
- **Omogenee:** i dati sono tutti dello stesso tipo
- **Non omogenee:** i dati sono di tipo diverso
- Esempio: il tipo di dato array rappresenta una struttura dati lineare, omogenea, a dimensione fissa

7

Strutture dati

- **Nozione astratta** svincolata dalla concreta rappresentazione della struttura nel modello di calcolo
- **Implementazione** descrive il modo con il quale la struttura è memorizzata e viene gestita dal calcolatore
 - Ogni struttura dati ammette più implementazioni con un costo diverso
 - ◆ Spazio di memorizzazione
 - ◆ Tempo per l'esecuzione delle operazioni primitive sulla struttura

8

Tempo di esecuz. valutato su metodi pubblici

Implementazione di una struttura dati

- Per valutare l'efficienza di procedure che usano tipi di dato primitivi si prescinde dalle caratteristiche specifiche di una macchina e si assume un'organizzazione abbastanza generica:
 - i dati sono contenuti in memoria
 - la memoria è divisa in celle, tutte di ugual ampiezza, ognuna delle quali può contenere un dato elementare
 - si accede ad una cella specificandone l'indirizzo
 - il tempo di accesso ad una cella si assume costante

9

Strutture dati statiche: array

- **Array** = blocco di elementi dello stesso tipo
- **Implementazione di un array** A di n elementi di tipo primitivo (int, char, ...):
 - è memorizzato in n celle consecutive a partire da un indirizzo ind_A
 - il tempo di accesso ad un generico elemento i-esimo è uguale al tempo di accesso della cella di indirizzo $\text{ind}_A + i$, quindi è **costante** o $O(1)$
 - Il passaggio per valore di A ad una procedura ha costo $O(n)$

10

Strutture dati statiche: record

- **Record** (o tipo aggregato) = blocco di dati in cui elementi diversi possono appartenere a tipi diversi

- Struct del linguaggio C

```
Struct {  
    char Nome[25];  
    int Età;  
    float ValutazioneCapacità;  
} Impiegato;
```

11

X

Strutture dati statiche: record

- Implementazione di un record a k campi: è memorizzato in celle di memoria consecutive:
- se i campi del record sono tutti dello stesso tipo allora la rappresentazione in memoria è analoga a quella di un array di k elementi
- se i campi del record sono di tipo diverso allora ogni campo occuperà un numero di celle pari al numero necessario per rappresentare il tipo di dato di ogni campo

- L'accesso al campo di indice i del record ha un costo $O(1)$
- Il passaggio per valore di un record ad una procedura ha costo:

$$\sum_{j=1}^k \text{sizeof}(\text{campo}_j) \in \mathcal{O}(\max_j(\text{sizeof}(\text{campo}_j)))$$

12

X