

# ASD - Algoritmi e Strutture Dati

Elia Ronchetti

@ulerich

2022/2023

# Indice

<b>1</b>	<b>Introduzione algoritmi</b>	<b>3</b>
1.1	Che cos'è un algoritmo? . . . . .	3
1.2	Analisi di un algoritmo . . . . .	3
1.3	Regole sullo Pseudocodice . . . . .	4
1.4	Esame . . . . .	5
1.5	Definizioni di base . . . . .	5

# Capitolo 1

## Introduzione algoritmi

### 1.1 Che cos'è un algoritmo?

Un algoritmo è

- Una sequenza di istruzioni elementari
- Agisce su un input per produrre un output
- Risolve un problema computazionale

Un algoritmo deve essere corretto e efficiente.

**Corretto** Significa che deve funzionare per qualsiasi input valido

**Efficiente** Deve occupare il minor spazio possibile ed impiegare il minor tempo possibile.

L'efficienza di un algoritmo si misura in termini di spazio e tempo

### 1.2 Analisi di un algoritmo

Per analizzare l'efficienza di un algoritmo si calcola il numero di istruzioni eseguite, ma esso non è univoco, varia in base all'input ricevuto, è quindi necessario individuare il **caso migliore** e il **caso peggiore**, essi si analizzano a parità di dimensioni, per questo non dipendono da essa. Dire che il caso migliore è quando l'array è vuoto non ha senso ai fini dell'analisi.

Per avere un'idea dei tempi di esecuzione è necessario calcolare il **Caso Medio**

NON è la media tra caso peggiore e caso migliore!

## 1.3 Regole sullo Pseudocodice

Gli algoritmi saranno scritti in Pseudocodice secondo le seguenti regole

- Il codice sarà simil C/Java
- Cicli: for, while, do-while
- Condizioni: if, else
- Indentazione + begin/end
- Commenti /\*.....\*/
- Assegnamenti  $A = 5$ ,  $A := 5$ ,  $A \leftarrow 5$
- Test del valore  $A == 5$
- Variabili: locali
- Array  $A[i] \rightarrow i \rightarrow 1 \dots n$
- Dati sono considerati oggetti con attributi (come  $\text{length}(A)$  per gli array)
- Puntatori: liste dinamiche
- Funzioni/Procedure - I parametri sono passati per valore (non per indirizzo)

**Macchina RAM (Random Access Machine)** La macchina su cui verranno eseguiti gli algoritmi sarà considerata RAM e quindi con le seguenti Caratteristiche

- Memoria ad accesso diretto
- No limiti memoria
- Sistema monoprocesso

## 1.4 Esame

L'esame sarà uno scritto con esercizi e domande di teoria. I parziali sono tendenzialmente riservati al primo anno, ma è possibile scrivere una email al prof 2 settimane prima del parziale e chiedere di poterlo sostenere anche se si è di un altro anno, sarà a sua discrezione concedere o meno questa opportunità. Si possono recuperare i parziali, è possibile anche tentare un recupero per migliorare un voto già positivo, accettando il rischio di che se il voto preso nell'esame di recupero è minore di quello originale si dovrà accettare quel voto.

## 1.5 Definizioni di base

**Algoritmo Corretto** Un algoritmo si definisce corretto se per tutti gli input si ottiene il risultato desiderato, l'algoritmo è corretto solo se garantisce la correttezza del risultato.

**Algoritmo efficiente** Minor utilizzo di **Spazio** e **Tempo**.

### Determinare l'efficienza di un algoritmo

Il primo passo è determinare il numero di istruzioni eseguite dall'algoritmo dato che così facendo non dipende dalla potenza dell'hardware e dall'input.

### Operazioni valutazione algoritmo

1. Conto le istruzioni **eseguite**
2. Determinare T peggiore - T migliore - T medio (la media non è fra T peggiore e T migliore)

Il tempo non sarà una quantità in secondi, ma dipenderanno da un parametro  $n$   $T(n)$ .

Quando devo scegliere un algoritmo mi baso sulla funzione  $n$ , dato che al crescere dell'input la funzione crescerà in modo lineare, quadratico, cubico, ecc. e questo mi mostrerà come cresce il tempo in funzione di  $n$ .

A parità di  $n$  controllo il fattore moltiplicativo.

**Esempio** I polinomiali hanno tempi di esecuzione accettabili, mentre i tempi esponenziali sono intrattabili, il problema è che esistono algoritmi esatti, ma sono esponenziali, per questo sono inutili dato che non con grandi input non si fermano mai.

Determinare il **Caso peggiore** serve per capire quanto tempo devo aspettare al massimo, quindi dopo quanto tempo avrò sicuramente un risultato, il **Caso minore**, determina il tempo minimo che devo aspettare, il **Caso medio** determina mediamente quanto tempo devo aspettare (non è la media fra T peggiore e T migliore).