PROGRAMMAZIONE INFORMATICA

ALGORITMI E STRUTTURE DATI

ALGORITMO

La parola algoritmo ha origine nel Medio Oriente. Essa proviene dall'ultima parte del nome dello studioso persiano Abu Jàfar Mohammed Ibn Musa Al-Khowarizmi, il cui testo di aritmetica (825 d.C. circa) esercitò una profonda influenza nei secoli successivi.

Un algoritmo descrive un metodo di risoluzione di un problema in un numero finito di passi. E' un **elenco finito di istruzioni**, di passi, che devono essere eseguiti per arrivare alla soluzione finale del problema.

Gli algoritmi fanno uso di **dati di ingresso** e sono in grado di produrre dei **risultati** che costituiscono la soluzione del problema in questione.

PROPRIETÀ DEGLI ALGORITMI

COMPLETEZZA: capacità dell'algoritmo di trovare una soluzione per il problema per il quale è stato sviluppato, per tutte le possibili combinazioni di dati di ingresso.

NON AMBIGUITÀ: nella descrizione dell'algoritmo non vi sono punti in cui si deve fare un cosa ed il suo opposto, o comunque non si hanno conflitti nella produzione dei risultati.

FINITEZZA: l'algoritmo è composto da un numero finito di istruzioni e termini in un numero finito di passi.

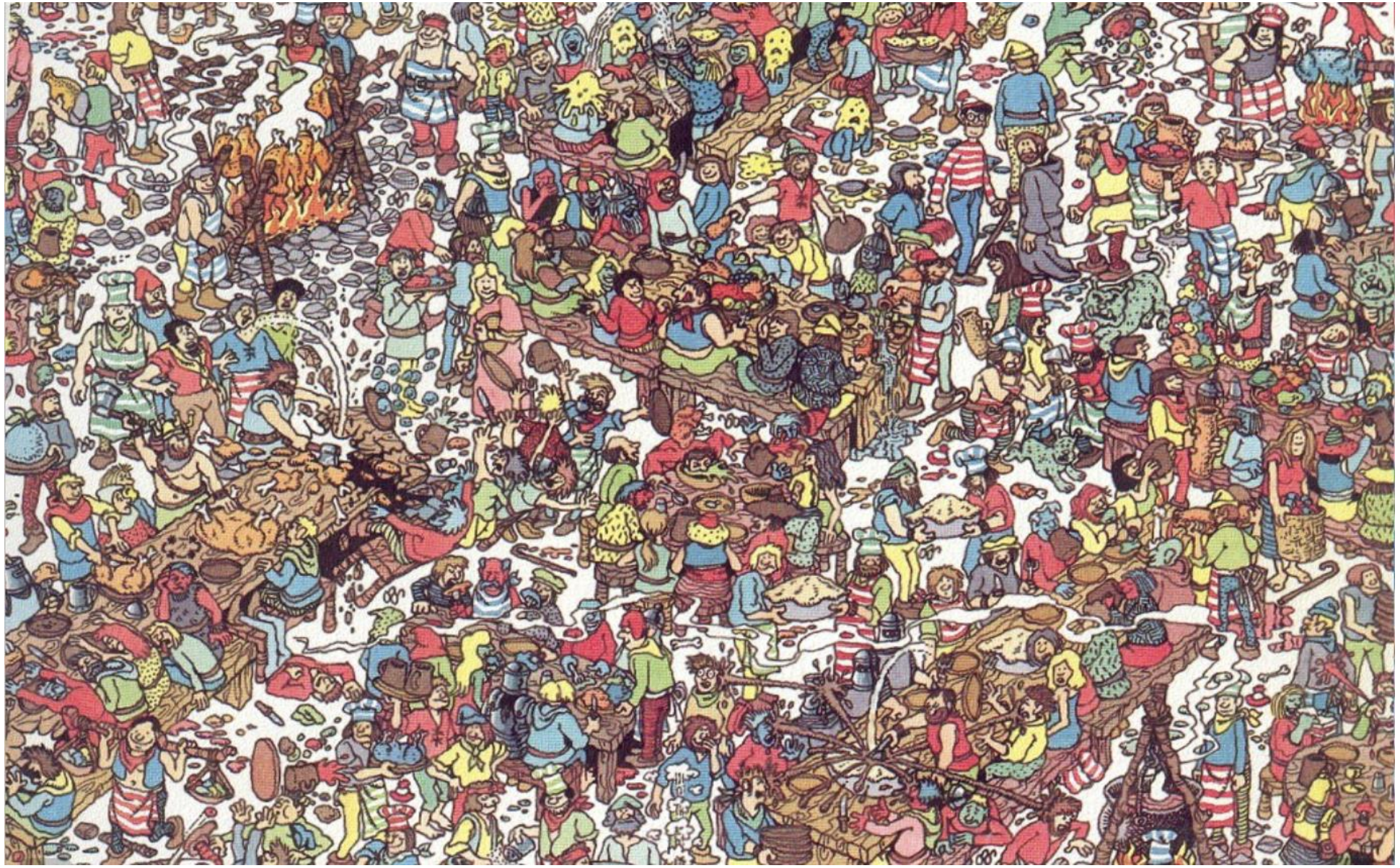
PROPRIETÀ DEGLI ALGORITMI

DETERMINISTICO: dato un particolare input, verrà prodotto sempre lo stesso output.

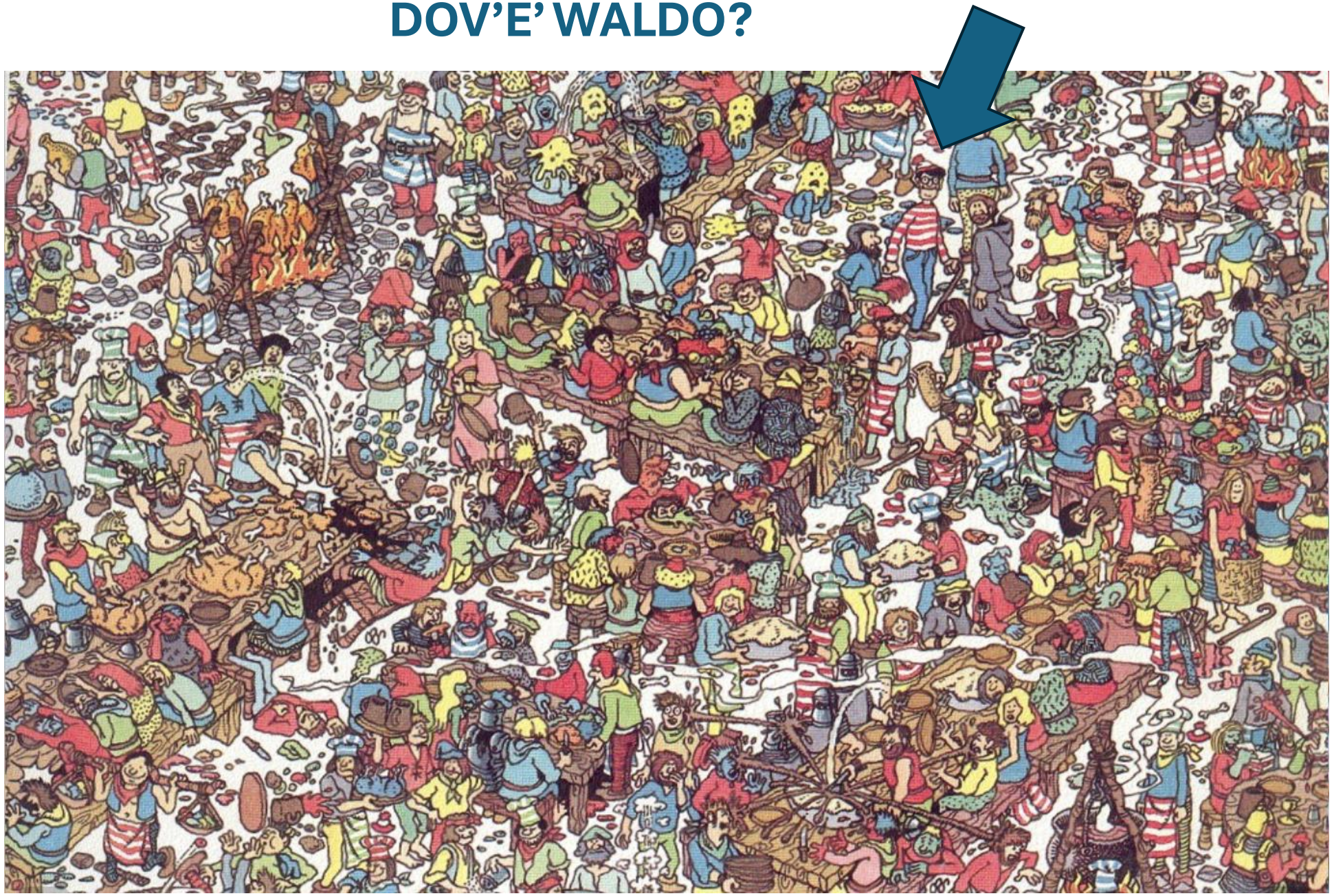
GENERALITÀ: L'algoritmo deve essere applicabile a una classe generale di problemi, non a uno specifico caso. Ciò significa che lo stesso algoritmo può essere usato con diverse istanze di input.

EFFICIENZA: capacità di un algoritmo di produrre risultati corretti nel minor numero possible di iterazioni. Viene misurata in base al tempo necessario per risolvere il problema e allo spazio di memoria occupato durante il calcolo.

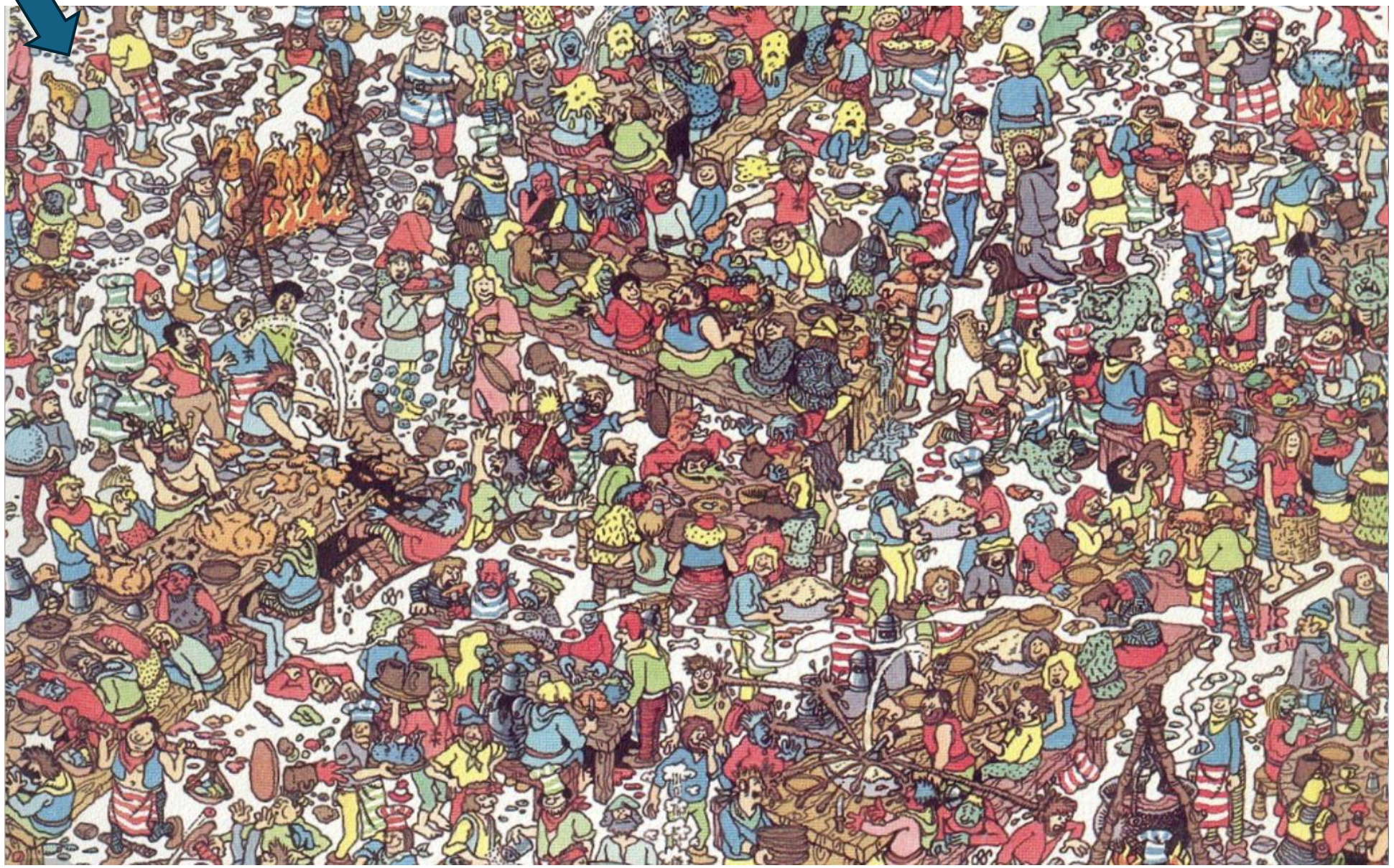
DOV'E' WALDO?



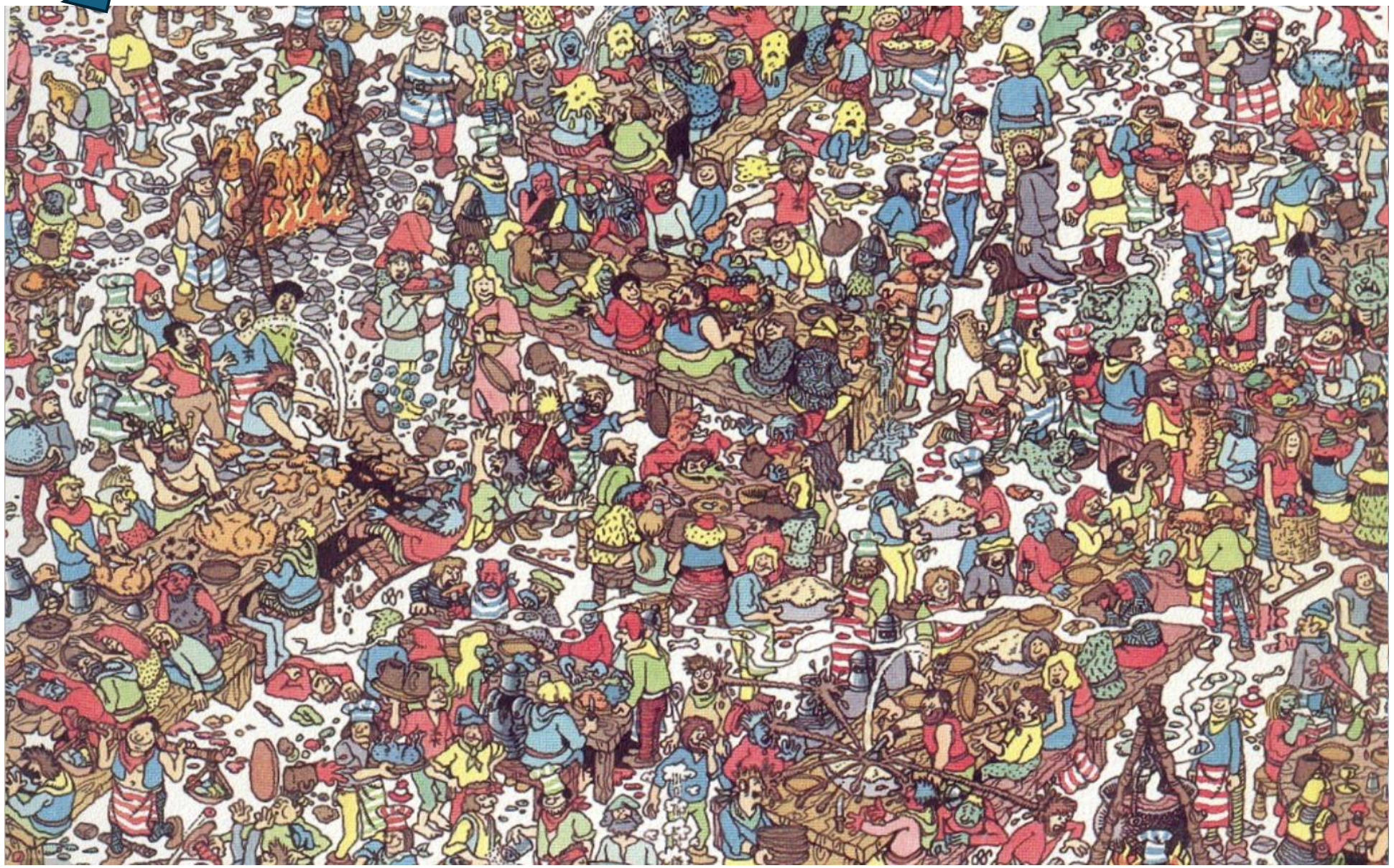
DOV'E' WALDO?



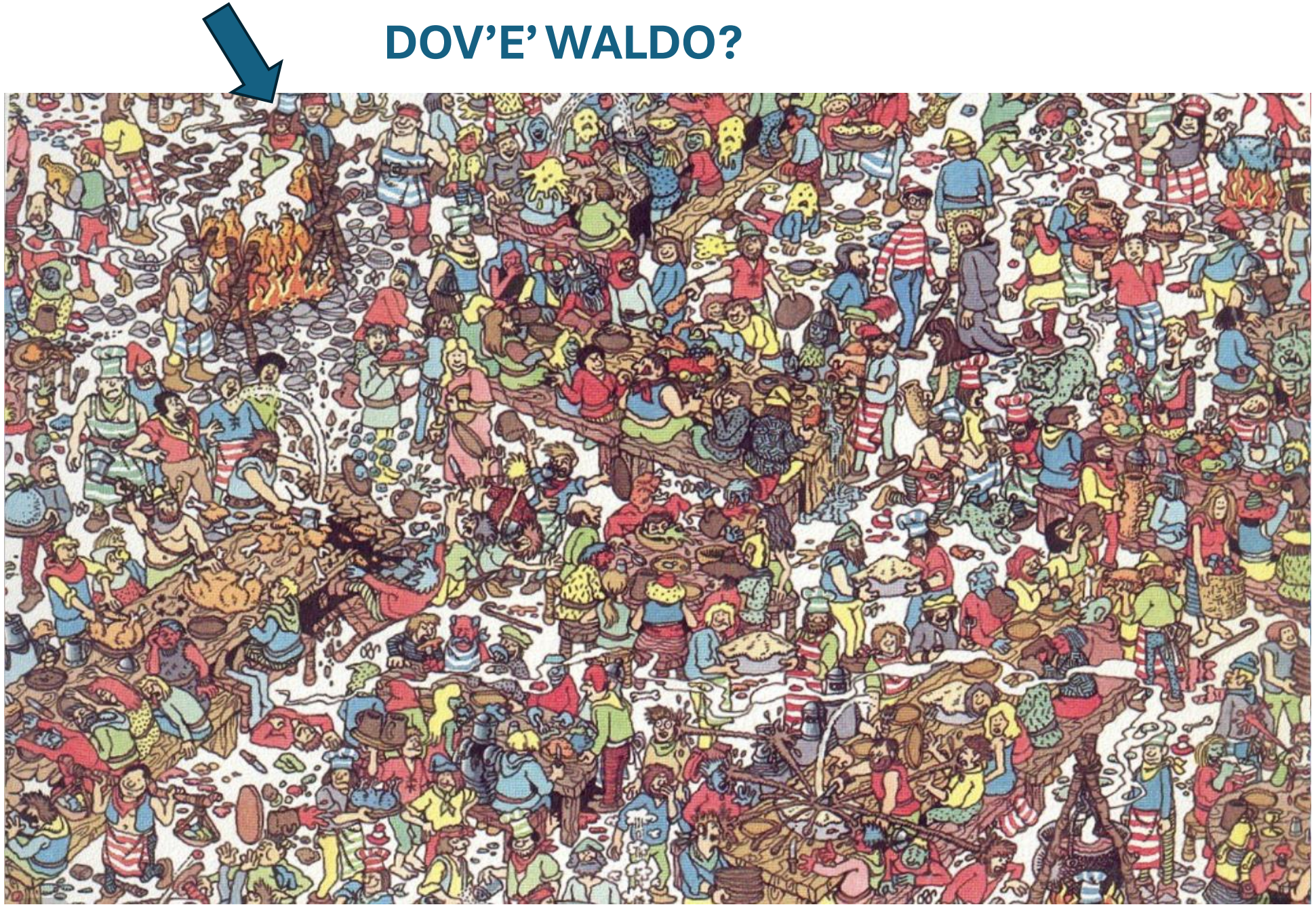
DOV'E' WALDO?



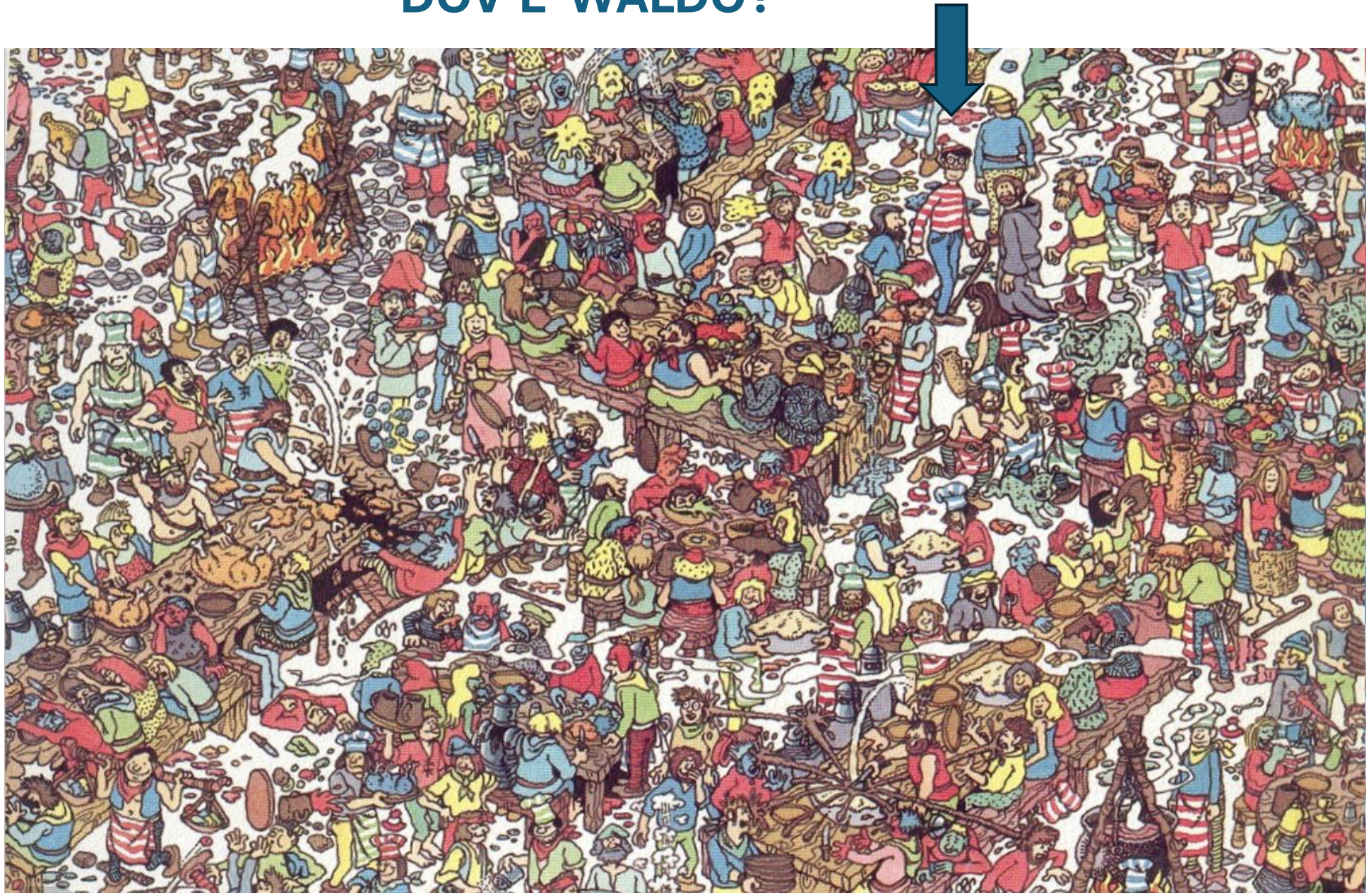
DOV'E' WALDO?



DOV'E' WALDO?



DOV'E' WALDO?



ORDINAMENTO

3	2	5	1	4

1	2	3	4	5

3 ⟷ 2	5	1	4

2	3 ⟷ 5	1	4

2	3	5 ⟷ 1	4

2	3	1	5 ⟷ 4

2 ⟷ 3	1	4	5

2	3 ⟷ 1	4	5

2	1	3 ⟷ 4	5

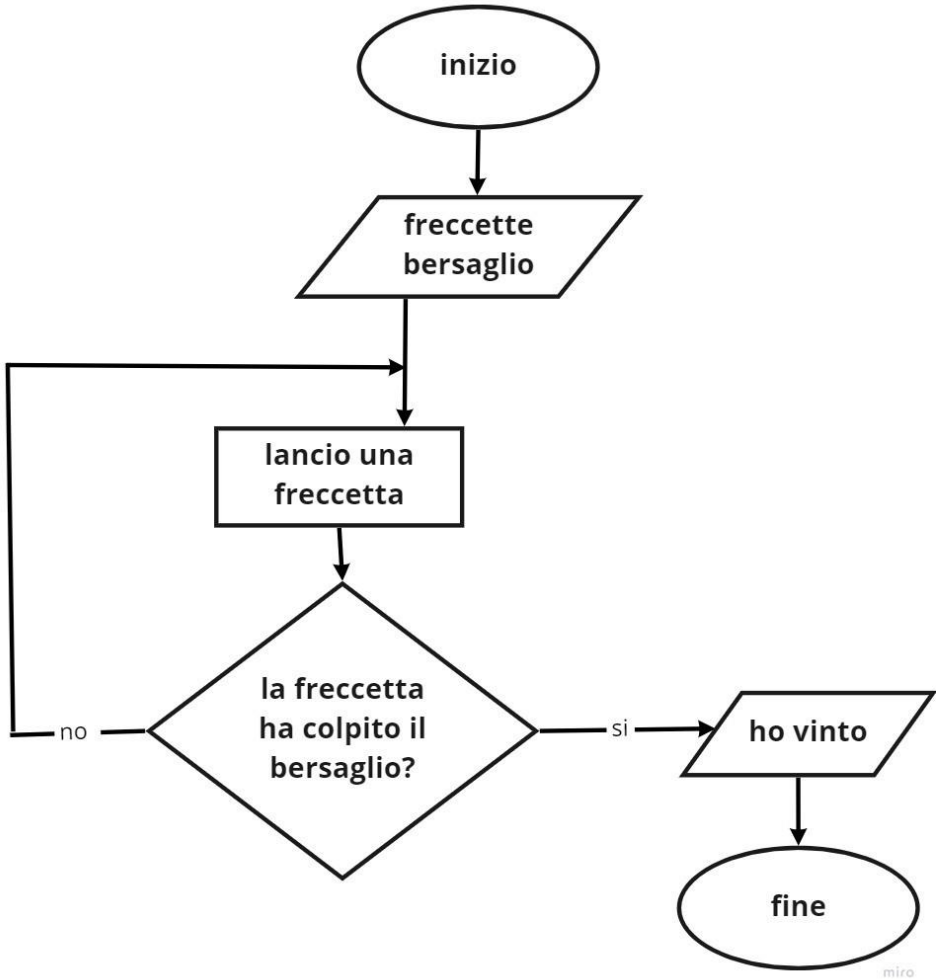
2 ⟷ 1	3	4	5

1	2	3	4	5

BUBBLE SORT

RAPPRESENTAZIONI DEGLI ALGORITMI

Diagrammi di flusso



Pseudocodice

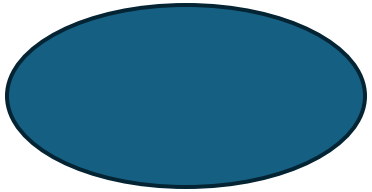
```
var freccette, bersaglio;

while(freccette !hit bersaglio) {
        lancia freccetta;
}

print("HO VINTO!");
```

DIAGRAMMI DI FLUSSO

Struttura costituita da blocchi connessi da frecce. Il flusso di lavoro è costituito dalle azioni contenute nei blocchi che vengono eseguite secondo l'ordine definito dalle frecce.

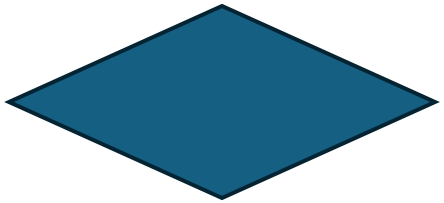
Ci sono 3 tipologie di blocchi:

- **Blocco terminale** che definisce l'inizio o la fine dell'algoritmo.



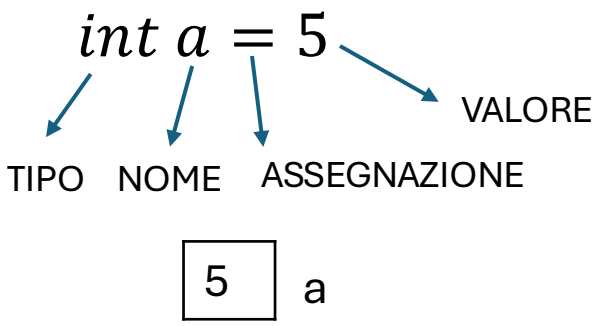
- **Blocco funzionale** che contiene un'istruzione elementare di tipo aritmetico, di assegnazione, di lettura o di scrittura.

- **Blocco decisionale** che contiene una condizione logica in base alla quale si decide quale delle due frecce uscenti seguire per la prosecuzione dell'esecuzione.

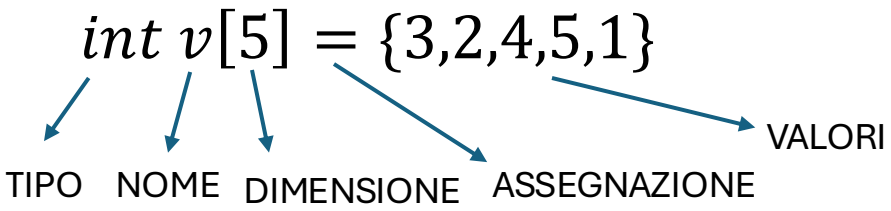


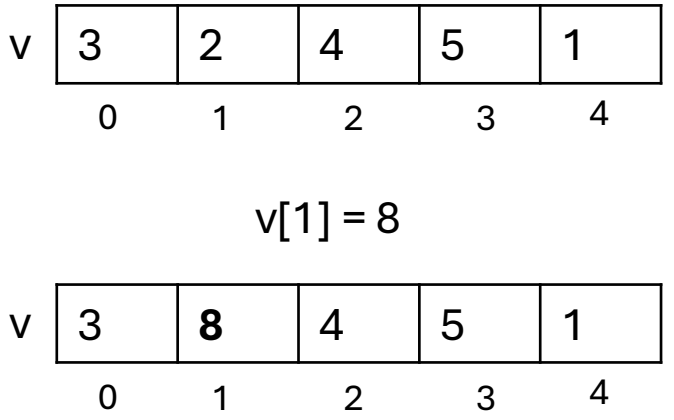
STRUTTURE DATO PRINCIPALI

- **VARIABILI:** a porzione di memoria (RAM) contenente un'informazione, identificata da un nome, un indirizzo ed un valore
  - **int** (numeri interi)
  - **float** (numeri reali)
  - **char** (caratteri)
  - **boolean** (vero/falso)



- **ARRAY:** sequenza ordinata di variabili omogenee.





STRUTTURE DATO PRINCIPALI

- **MATRICI:** array bidimensionale.

$$int\ m[3][4] = \{\{3,2,4,6\},\{5,8,9,7\},\{3,7,5,9\}\}$$

TIPO — NOME — NUMERO RIGHE — NUMERO COLONNE — VALORI RIGA 1 — VALORI RIGA 2 — VALORI RIGA 3

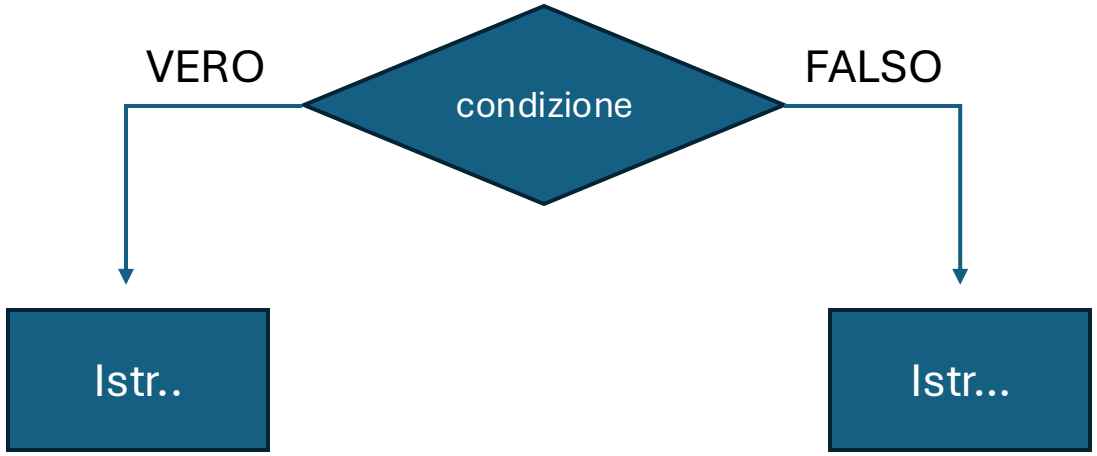
m	0	1	2	3
0	3	2	4	6
1	5	8	9	7
2	3	7	5	9

m[1][2] = 11

m	0	1	2	3
0	3	2	4	6
1	5	8	**11**	7
2	3	7	5	9

STRUTTURE DI CONTROLLO

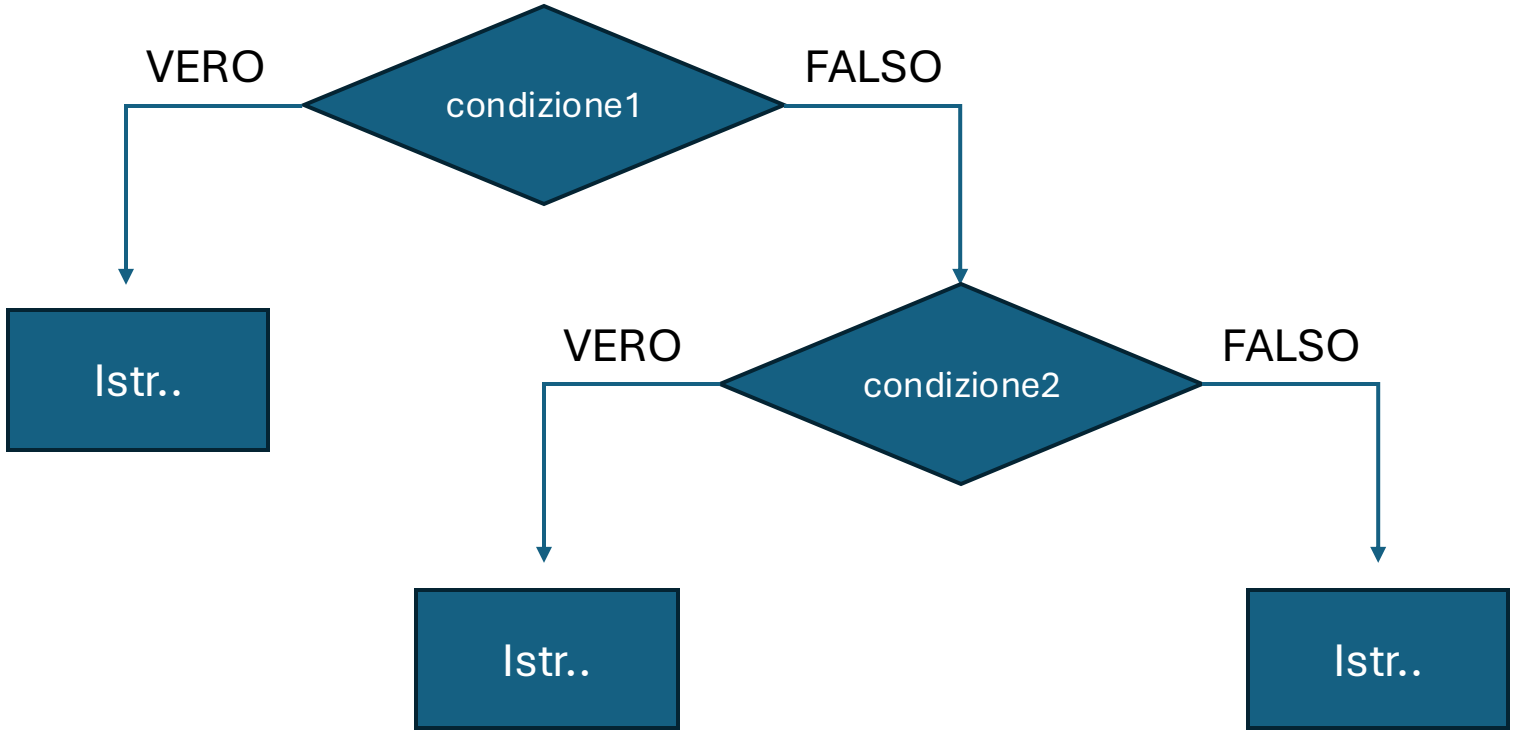
- **IF/ELSE:** struttura di controllo condizionale utilizzata per eseguire blocchi di codice in base a una condizione booleana (vero o falso).




```
if(condizione) {
        istr...
} else {
        istr...
}
```

STRUTTURE DI CONTROLLO

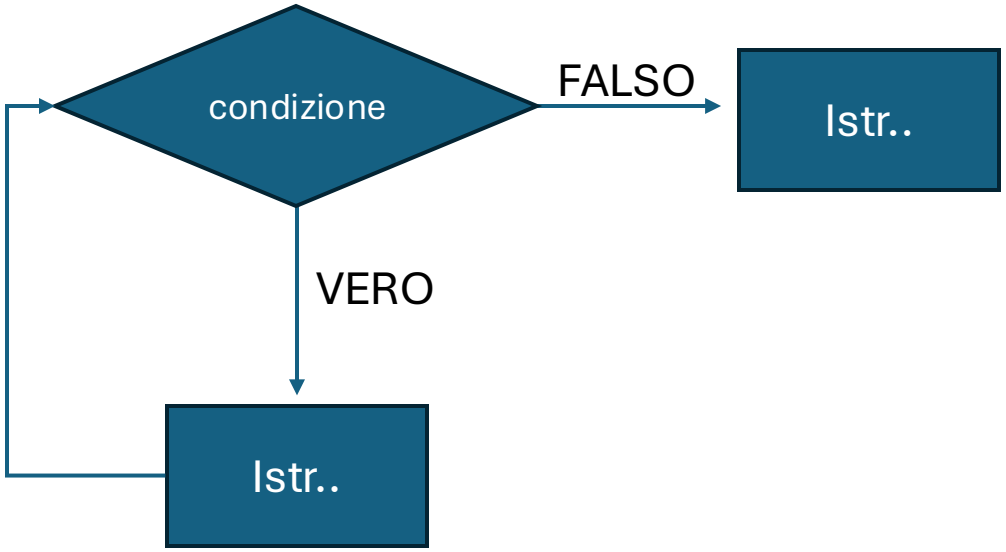
- **IF/ELSE:** struttura di controllo condizionale utilizzata per eseguire blocchi di codice in base a una condizione booleana (vero o falso).




```
if(condizione1) {
        istr...
} else if(condizione2) {
        istr...
} else {
        istr...
}
```

STRUTTURE DI CONTROLLO

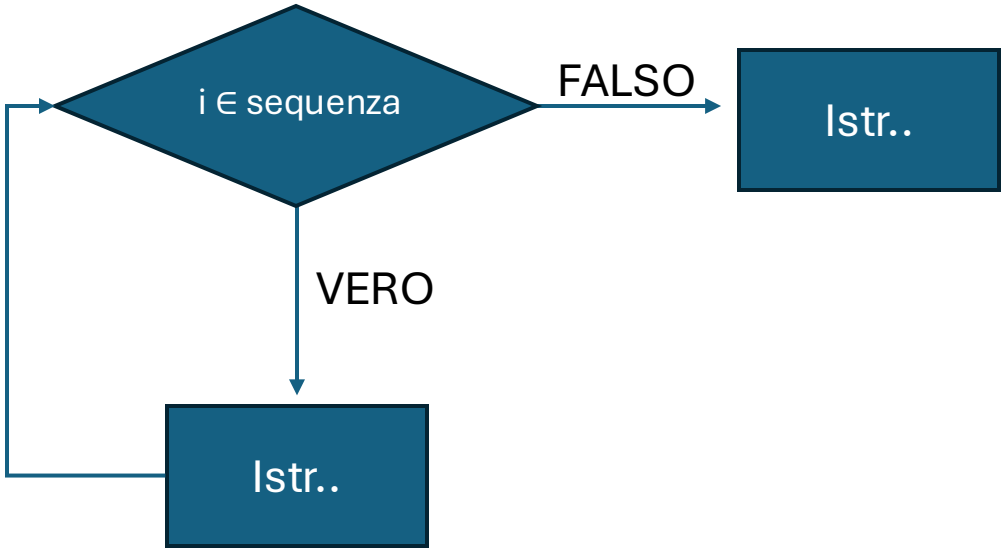
- **CICLO:** struttura di controllo che permette di ripetere l'esecuzione di un blocco di codice finché una determinata condizione è vera o per un numero specificato di volte.




```
while(condizione) {
        istr..
}
```

STRUTTURE DI CONTROLLO

- **CICLO:** struttura di controllo che permette di ripetere l'esecuzione di un blocco di codice finché una determinata condizione è vera o per un numero specificato di volte.



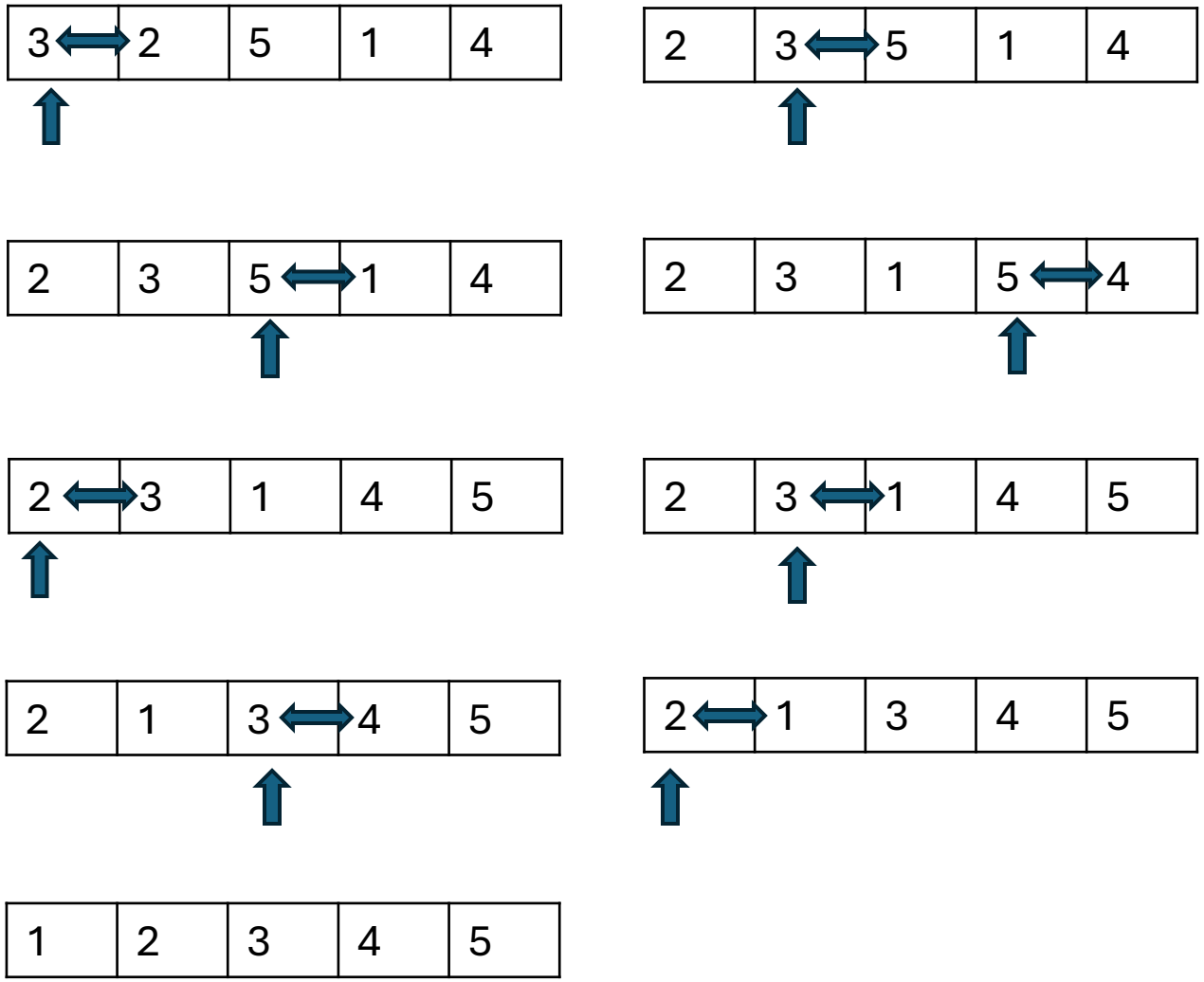

```
for(i in sequenza) {
        istr..
}
```

STRUTTURE DI CONTROLLO

- **FUNZIONE**: blocco di codice che svolge una specifica operazione, definito una sola volta e poi riutilizzabile in diverse parti del programma. Le funzioni aiutano a organizzare il codice, rendendolo più modulare, leggibile e mantenibile.

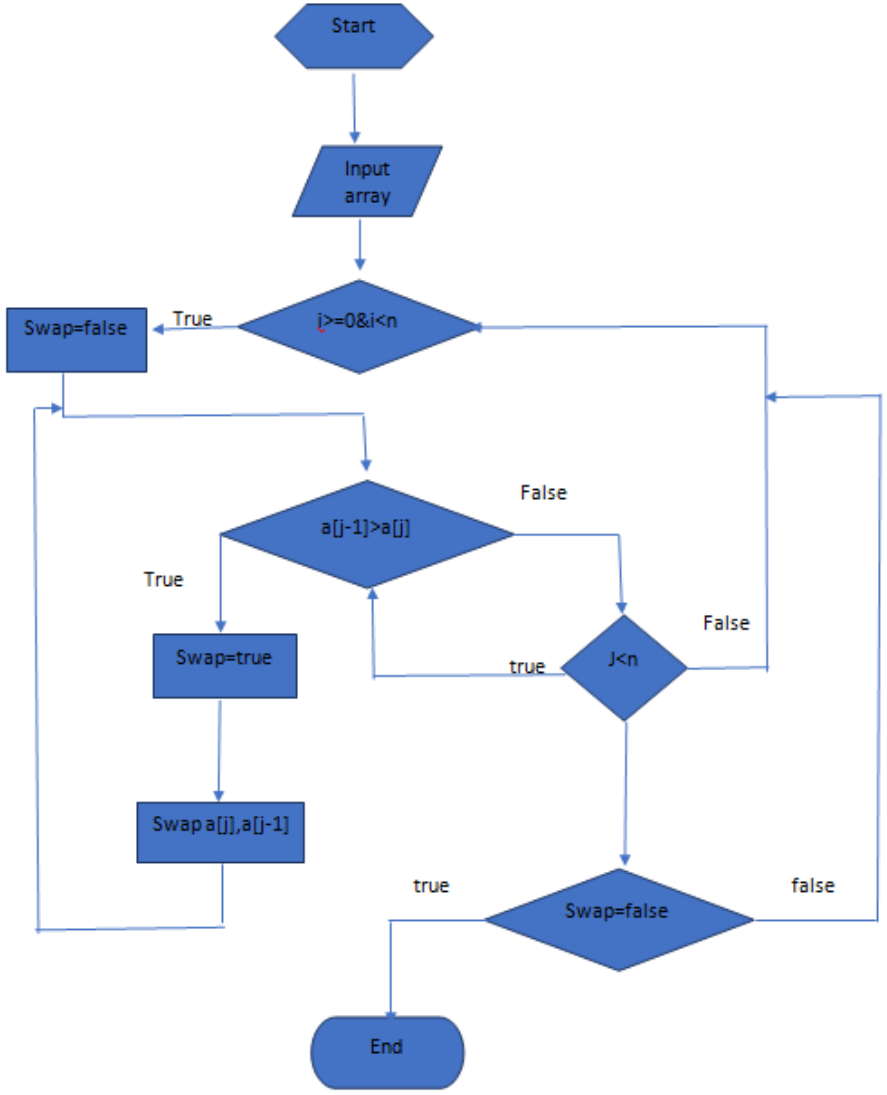
```
def nome_funzione(parametri) {
        istr..
}
```

BUBBLE SORT



BUBBLE SORT

```
int v[5] = {3,2,5,1,4}, x , j, i
boolean scambio

for(i in seq(0,4)) {
        scambio = False
        for(j in  seq(0,(4-i))) {
                if(v[j] > v[j+1]) {
                        x = v[j+1]
                        v[j+1] = v[j]
                        v[j] = x
                        scambio = True
                }
        }
        if(!scambio) {
                break
        }
}
```



BUBBLE SORT

```
def bubble(v, n) {
        int x,j,l
        for(i in seq(0,n)) {
                scambio = False
                for(j in  seq(0,(n-(i+1)))) {
                        if(v[j] > v[j+1]) {
                                x = v[j+1]
                                v[j+1] = v[j]
                                v[j] = x
                                scambio = True
                        }
                }
                if(!scambio) {
                        break
                }
        }
        return(v)
}
```