ARRAY
UN MONDO SCONTATO

1	3	20	31	43	52

Ma è tutto qui?

Bacaro
Tech

CODE AND FUN

Buona sera a tutti.

Siamo Giorgio e Michele e vi presentiamo il nostro, progetto personale.

BACARO TECH

Bacaro Tech

CODE AND FUN

Bacaro
Tech

CODE AND FUN

DOVE NASCE BACARO TECH?

CICHETI + VINO + PERSONE = BACARO

BUG + CODING + PERSONE = TALK TECH

Sono degli eventi, a tema Tecnologico - Informatico, con l'intento di passare una piacevole serata ricca di spunti tecnologici.

Noi ci mettiamo il massimo,
per bug fix e code review
Ci vediamo in Osteria!

Bacaro Tech

CODE AND FUN

CHE COSA VOGLIAMO FARE CON BACARO TECH

DIVULGAZIONE

Vogliamo portare a tutte le seniority il coding e ciò che ci orbita intorno.

WORKSHOP ⟵

⟶ **TALK TEMATICI**

ESERCIZI DI CODING

E MOLTO ALTRO

Bacaro
Tech

CODE AND FUN

COSA VI VIENE IN MENTE SE DICO ARRAY?

Bacaro
Tech

CODE AND FUN

CHE LINGUAGGI SONO?

```python
miei_numeri = [10, 20, 30, 40, 50]
print(miei_numeri[2])  # Stampa '30'
```

```java
@GetMapping
public String[] getWineList() {
    String[] wineList = new String[]{"Asti", "Chianti", "Marsala", "Prosecco"};
    return wineList;
}
```

```go
primes := [6]int{2, 3, 5, 7, 11, 13}
```

```python
frutta = ["mela", "banana", "arancia"]
print(frutta[1])  # Stampa 'banana'
```

```cpp
string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};
```

```typescript
let a: number[] = [0,1,2,4]
```

```php
$array = [
    "foo" => "bar",
    "bar" => "foo",
];
```

Bacaro
Tech

CODE AND FUN

OGGI COSA VEDIAMO?

ARRAY ALLA BASE
CARATTERISTICHE DEGLI ARRAY
COMPLESSITÀ
RICORSIONE

ALGORITMI DI RICERCA
ALGORITMI DI ORDINAMENTO

Bacaro
Tech

CODE AND FUN

PARTIAMO DA UNA DOMANDA CHE COS'È UN ARRAY?

Bacaro
Tech

CODE AND FUN

CHE COS'È UN ARRAY

Un array è una collezione **ordinata** di elementi dello **stesso tipo**, anche se non sembra.

Bacaro
Tech

CODE AND FUN

UN ARRAY CHE COS'É

Possiamo schematizzare un array come un'insieme di valori ai quali sono posti degli indici per accedere ai valori contenuti

ELEMENTI
INDICI

1	3	20	31	43
0	**1**	2	3	4

Le posizioni vanno da 0 a numero di elementi - 1!*

Bacaro
Tech

CODE AND FUN

CARATTERISCHE DEGLI ARRAY

Dimensione Fissa o Dinamica: Gli array possono avere dimensione stabilita o modificabile durante l'esecuzione del programma.

Ordine: Gli elementi in un array seguono un ordine specifico, determinato da indici, solitamente a partire da zero(esiste mathlab).

Accesso Diretto: Gli array consentono di raggiungere direttamente un elemento conoscendo il suo indice.

Elementi Omogenei: Tutti gli elementi in un array sono dello stesso tipo di dato, anche se all’occhio non sembra.

Memoria Contigua: Gli elementi sono conservati consecutivamente in memoria, permettendo l’accesso diretto e performance migliori

Bacaro
Tech

CODE AND FUN

QUANDO SI USA UN ARRAY?

```
//SENZA ARRAY
let item1 = "pane";
let item2 = "latte";
let item3 = "uova";
console.log(item1, item2, item3);

//UTILIZZANDO GLI ARRAY
let shoppingList: string[] = ["pane", "latte", "uova"];
console.log(shoppingList);
```

Utilizzando gli array è possibile evitare queste sbavature di codice!
-clean code

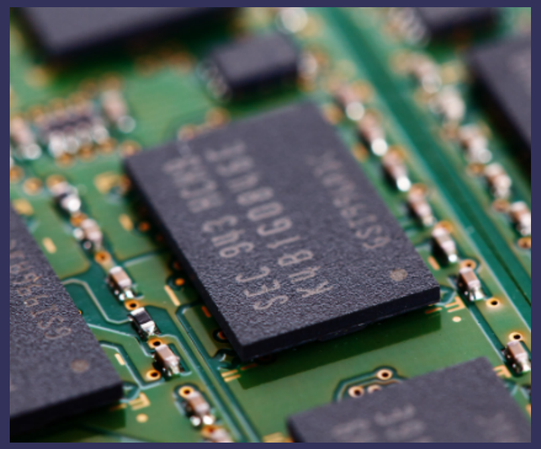
Bacaro
Tech

CODE AND FUN

SE VI DICO MEMORIA COSA PENSATE?

Bacaro
Tech

CODE AND FUN



Bacaro
Tech

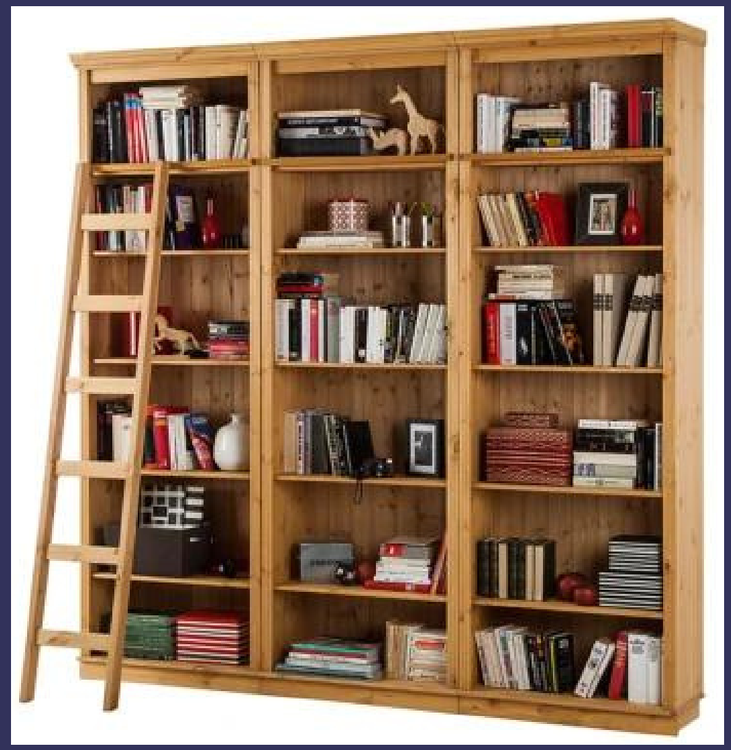
CODE AND FUN

FACCIAMO UN SALTO ASTRATTO NELLA MEMORIA

Bacaro
Tech

CODE AND FUN

FACCIAMO UN SALTO ASTRATTO NELLA MEMORIA



Bacaro Tech

CODE AND FUN

MEMORIA: LA LIBRERIA DELLA CPU

Pensiamo alla **memoria** coma a un armadio, una libreria o a una cassettiera

CASSETTI / Grucce / Ripiani = Celle di Memoria. Ogni posto contiene un oggetto.

La posizione dell'oggetto nel contenitore = INDIRIZZO o Riferimento

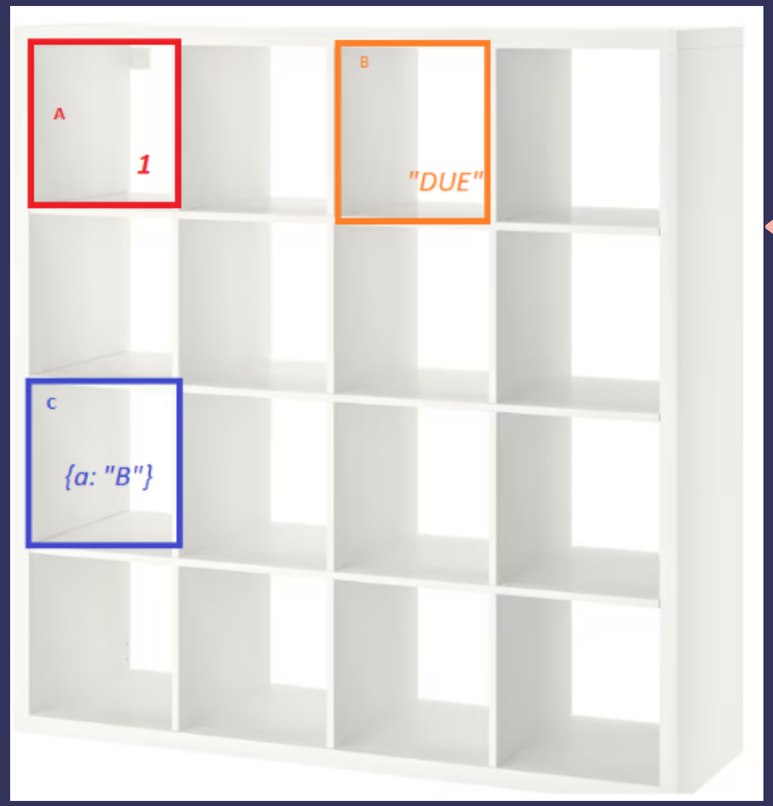
"Leggi tutti i libri nella sezione 'Dizionari'"

"Prendi il calzino che si trova sull'armadio di destra nel terzo cassetto"

Bacaro
Tech

CODE AND FUN

VARIABILI NELLA LIBRERIA DELLA CPU



Rappresentazione

```
var A = 1;
var B = "DUE";
var C = { a: "B" };
```

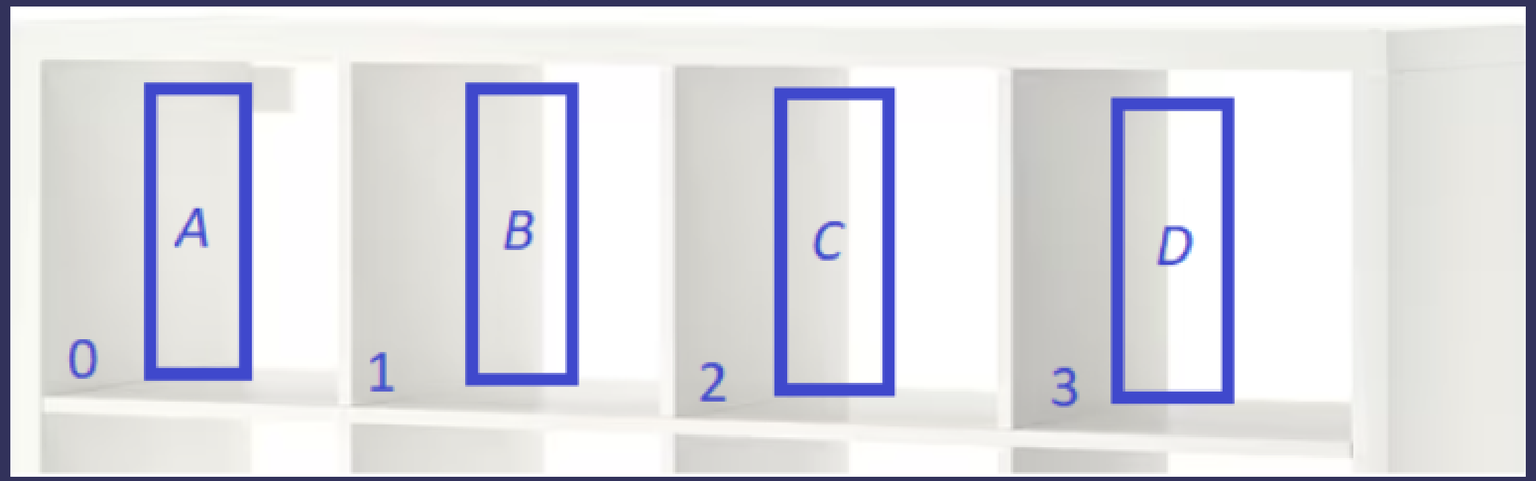
Bacaro
Tech

CODE AND FUN

ARRAY NELLA LIBRERIA DELLA CPU

```
var myArray = [A, B, C, D]
```

Rappresentazione

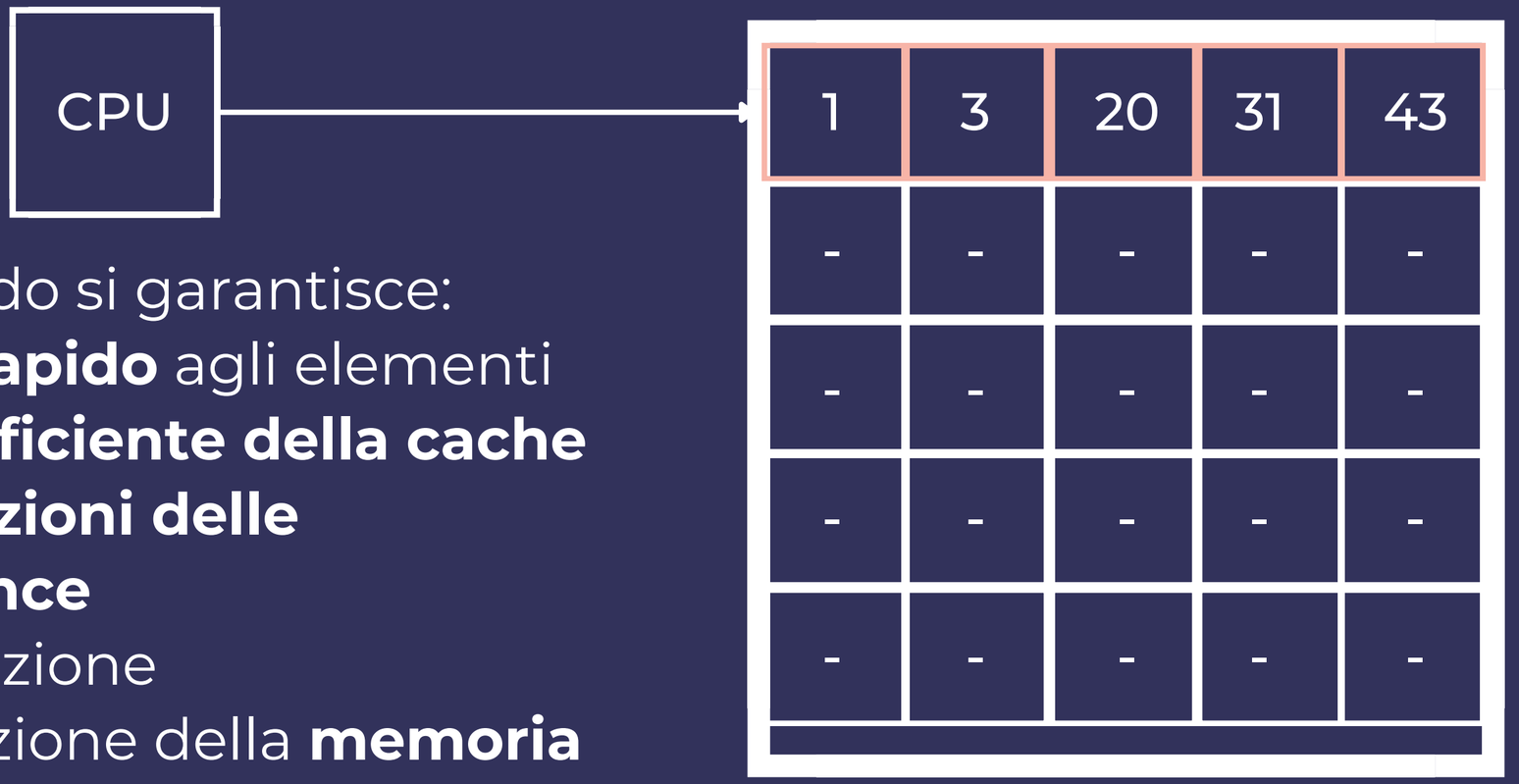


Bacaro
Tech

CODE AND FUN

ARRAY E MEMORIA

Grazie alla contiguità delle celle in memoria che formano l'array, la CPU può eseguire i calcoli con maggiore efficienza.



In questo modo si garantisce:

1. **Accesso rapido** agli elementi
2. **Utilizzo efficiente della cache**
3. **Ottimizzazioni delle performance**
4. Semplificazione dell'allocazione della **memoria dinamica**

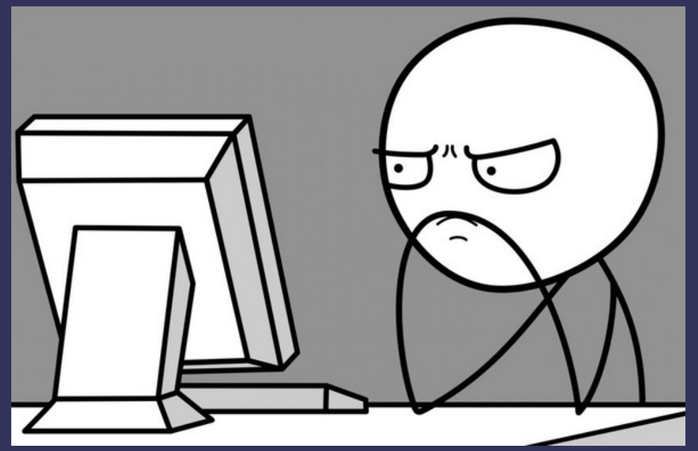
Bacaro
Tech

CODE AND FUN

ARRAY NEI LINGUAGGI

IT’S TIME TO CODING...

... ma servono ancora degli strumenti



Bacaro
Tech

CODE AND FUN

CODING CON APPROCCIO GENERICO

Per ora, non ha senso andare a vedere ogni linguaggio come definisce e come manipola un array.

Il nostro obbiettivo é di **imparare a usare gli array in modo generico**, indipendentemente dalla tecnologia che “ci sta sotto”.

Il **vantaggio** é che potrete approcciare gli algoritmi e i concetti con il linguaggio che più vi é comodo e chissà, magari sperimentare nuove tecnologie.

Bacaro
Tech

CODE AND FUN

STRUMENTI PER POTER GENERALIZZARE

ASTRAZIONE

PSEUDOCODICE

PRATICA!
(ANCA MASSA)

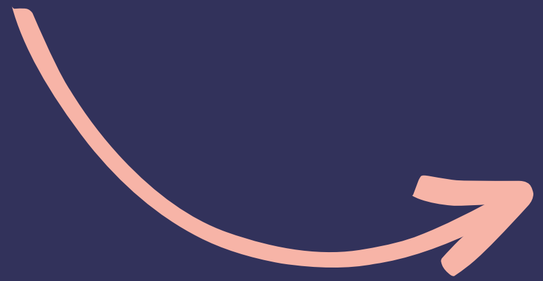
Bacaro
Tech

CODE AND FUN

ESEMPIO DI CODICE E PSEUDO CODICE MESSO A CONTRONTO

```
//SENZA ARRAY
let item1 = "pane";
let item2 = "latte";
let item3 = "uova";
console.log(item1, item2, item3);

//UTILIZZANDO GLI ARRAY
let shoppingList: string[] = ["pane", "latte", "uova"];
console.log(shoppingList);
```



```
// SENZA ARRAY

VARIABILE item1 COME STRINGA = "pane"

VARIABILE item2 COME STRINGA = "latte"

VARIABILE item3 COME STRINGA = "uova"



// UTILIZZANDO

VARIABILE shoppingList COME ARRAY DI STRINGHE

INSERISCI "pane" IN shoppingList

INSERISCI "latte" IN shoppingList

INSERISCI "uova" IN shoppingList
```

Bacaro
Tech

CODE AND FUN

COME FACCIAMO A DIRE CHE UN ALGORITMO È MEGLIO DI UN ALTRO?

Bacaro Tech

CODE AND FUN

MATEMATICA E PROGRAMMAZIONE: O GRANDE E COMPLESSITÀ COS'È?

Complessità temporale

Questo tipo di complessità si riferisce al **tempo necessario** per eseguire un algoritmo.

Quanto tempo ci vuole per risolvere un problema?

Complessità spaziale

Questo tipo di complessità si riferisce alla **quantità di memoria necessaria** per eseguire un algoritmo.

Quanto spazio viene utilizzato per memorizzare dati temporanei?

Bacaro
Tech

CODE AND FUN

MATEMATICA E PROGRAMMAZIONE: O GRANDE E LA COMPLESSITÀ COSA SONO?

La notazione "O grande" è una notazione utilizzata in informatica e nell'analisi degli algoritmi per **descrivere la complessità** temporale o spaziale di un algoritmo.

La notazione "O grande" aiuta a classificare gli algoritmi in base a quanto cresce il loro consumo di risorse (tempo o spazio) in relazione all'input.

Facciamo un esempio concreto per semplificare questo concetto!

Bacaro
Tech

CODE AND FUN

CONCETTO DI COMPLESSITÀ IN UN ESEMPIO REALE

Rappresentate 16 rettangoli
in un foglio.

Come lo faresti?

Bacaro Tech

CODE AND FUN

CONCETTO DI COMPLESSITÀ IN UN ESEMPIO REALE

Rappresentate 16 rettangoli in un foglio

Primo modo: "Disegna 16 rettangoli su un foglio."
T: O(16) == O(n).

Secondo modo: "Piegare il foglio 4 volte."
T: O(log 16) == O(log n).

Ci sono molti modi di fare una cosa, ma questo non significa che non ci siano dei modi migliori e peggiori di farla!

Bacaro
Tech

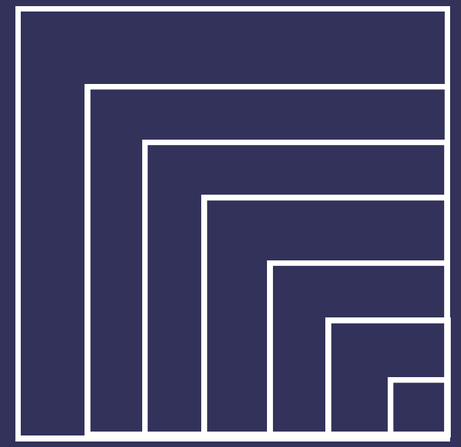
CODE AND FUN

Bacaro
Tech

CODE AND FUN

RICORSIONE CHE COS‘È?

La **ricorsione** è un concetto chiave in informatica. Essa si riferisce alla capacità di una funzione o di un algoritmo di richiamare se stesso per risolvere un problema più grande o complesso, **riducendo l’input**.



La ricorsione è una tecnica potente e flessibile utilizzata in molte aree dell'informatica, e ogni funzione ricorsiva ha 2 “pezzi” che la caratterizzano:

1. **Caso base**, dove termina la ricorsione
2. **Passo Ricorsivo**, dove continua la ricorsione

Bacaro
Tech

CODE AND FUN

UN ESEMPIO DI RICORSIONE

```
funzione fattoriale(n):
    se n è uguale a 1:
        restituisci 1 (caso base)
    altrimenti:
        calcola il fattoriale di n-1 in modo ricorsivo
        moltiplica il risultato per n e restituisci il risultato
```

Funzione ricorsiva per il calcolo del fattoriale di n

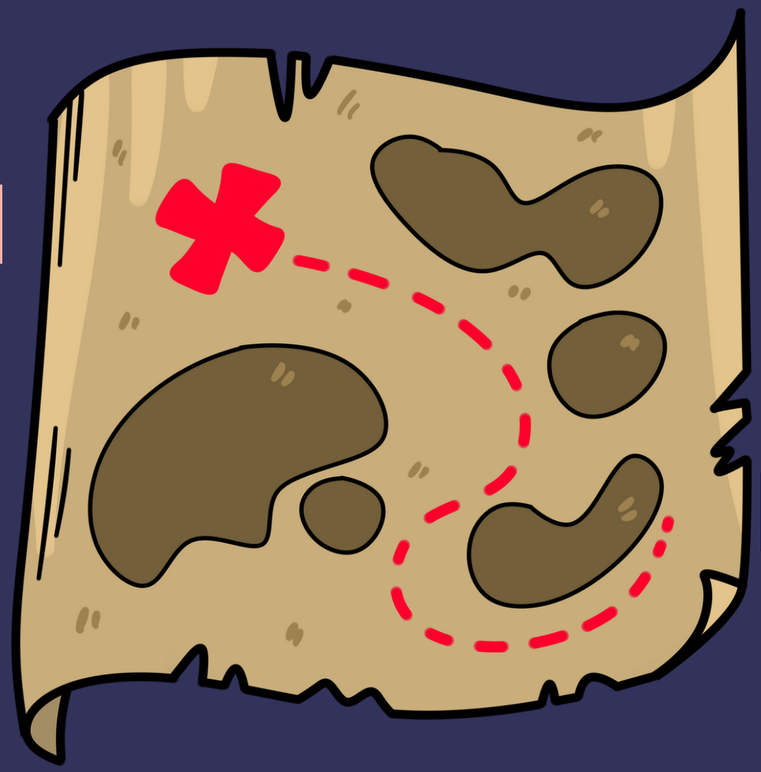
Quando chiami questa funzione con un valore n, essa utilizzerà la ricorsione per calcolare il fattoriale di n.

La funzione si chiama continuamente con valori più piccoli di n fino a raggiungere il caso base e quindi calcola il risultato finale.

Bacaro
Tech

CODE AND FUN

ALGORITMI



INIZIO TALK

Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCA LINEARE

Struttura dei dati: La ricerca lineare può essere applicata senza nessuna precodizione.

Metodo di ricerca: La ricerca lineare procede sequenzialmente attraverso la sequenza dei dati dall'inizio alla fine o viceversa, e ogni elemento viene confrontato uno per uno con l'elemento cercato.

Complessità temporale: O(n) minimo*

Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCA LINEARE

Bacaro
Tech

CODE AND FUN

RICERCA LINEARE
ESEMPIO DI CODICE

```
funzione ricercaLineare(array, elemento):
    per ogni elemento nell'array:
        se elemento è uguale all'elemento corrente:
            restituisci l'indice dell'elemento corrente
    restituisci -1 (se l'elemento non è presente)
```

Statistiche:

- tempo esecuzione medio O(n)
- tempo esecuzione nel caso peggiore O(n)
- spazio di occupazione O(1)

Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCA BINARIA

Struttura dei dati: La ricerca binaria richiede che i dati siano ordinati.

Metodo di ricerca: La ricerca binaria divide ripetutamente la sequenza in due parti uguali e confronta l'elemento desiderato con l'elemento centrale della sequenza, e in base al confronto, l'algoritmo decide se continuare a cercare nella metà superiore o inferiore della sequenza.
Questo processo si ripete fino a quando l'elemento è stato trovato o fino a quando si determina che l'elemento non è presente.

Complessità temporale: log(n)

Bacaro
Tech

CODE AND FUN

RICERCA BINARIA
ESEMPIO DI CODICE

```
funzione ricercaBinaria(array, elemento):
    inizio = 0
    fine = lunghezza dell'array - 1

    mentre inizio <= fine:
        medio = (inizio + fine) / 2
        se array[medio] è uguale all'elemento:
            restituisci medio
        se array[medio] < elemento:
            inizio = medio + 1
        altrimenti:
            fine = medio - 1

    restituisci -1 (se l'elemento non è presente)
```

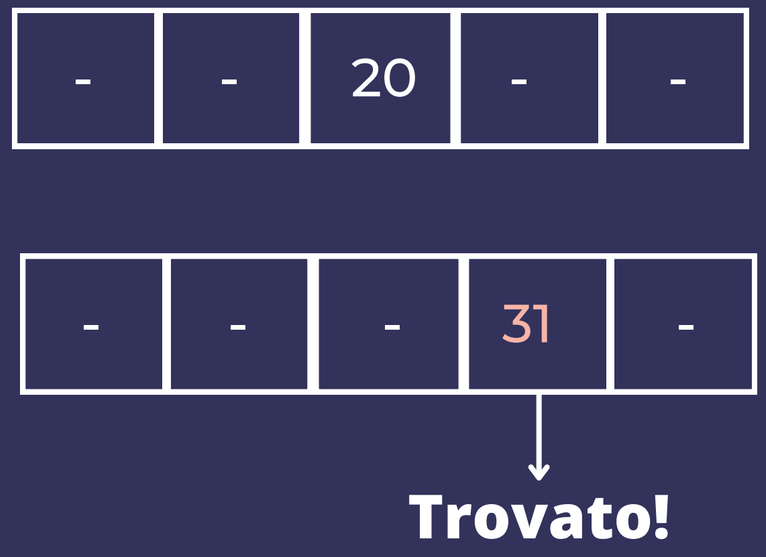
Statistiche:

- tempo esecuzione medio O(logn)
- tempo esecuzione nel caso peggiore O(logn)
- spazio di occupazione O(1)

Bacaro
Tech

CODE AND FUN

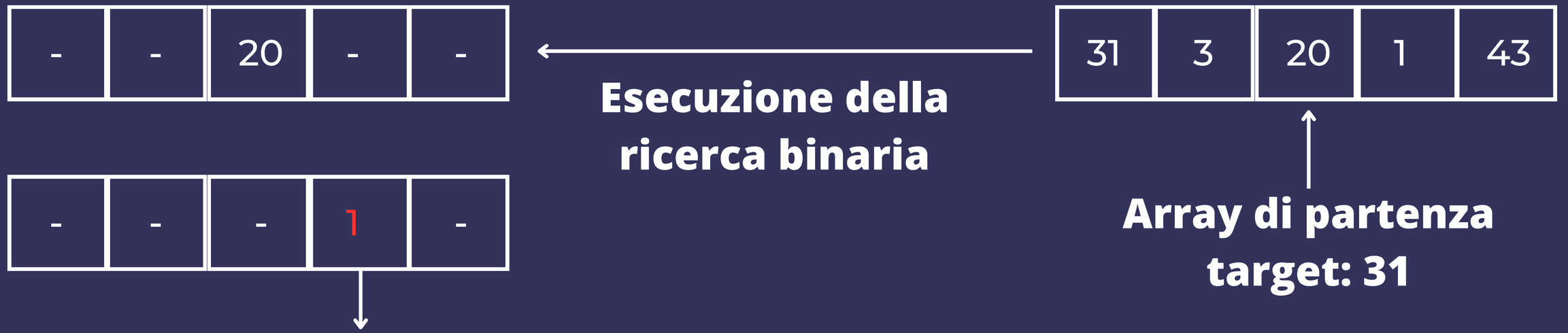
ARRAY E ALGORITMI
RICERCA LINEARE



Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCA LINEARE



Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCE A CONFRONTO

Ricerca Sequenziale

Tipo: Iterativo.
Array: Non ordinato.
Complessità: O(n).
Operazione: Confronto lineare degli elementi fino al ritrovamento.

Ricerca Binaria(BS)

Tipo: "Dividi e conquista".
Array: Ordinato.
Complessità: O(log n).
Operazione: Divide l'array in metà ripetutamente fino al ritrovamento.

MA...

Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCE A CONFRONTO

Visto che il binary funziona
solo se l’array é ordinato,
non mi conviene usare
sempre la ricerca lineare?

Bacaro
Tech

CODE AND FUN

ARRAY E ALGORITMI
RICERCE A CONFRONTO

Complichiamo il problema:
Si vuole trovare il 50 e fare in modo che esso sia l'unico nell'array. Non si può modificare l'array che viene fornito.

Array ordinato:

1	1	2	33	33	50	50	50	234	1000

Array non ordinato:

1	50	1000	33	234	2	50	33	50	1

Bacaro
Tech

CODE AND FUN

L'ARRAY ORDINATO É SEMPRE DESIDERATO ANCHE SE ESPLICITAMENTE NON RICHIESTO!

Bacaro
Tech

CODE AND FUN

COSA SONO GLI ALGORITMI DI ORDINAMENTO

Gli algoritmi di ordinamento sono procedure o regole ben definite per organizzare un insieme di dati in un ordine specifico, rispettando un certo criterio, come l'ordine crescente o decrescente, in modo da semplificare la ricerca, il recupero e altre operazioni sulle dati

Bubble sort

I sort che ne fanno parte possono si basare sia sul confronto che su altre tecniche!

Quick sort

Bacaro
Tech

CODE AND FUN

CHE COS‘È IL BUBBLE SORT

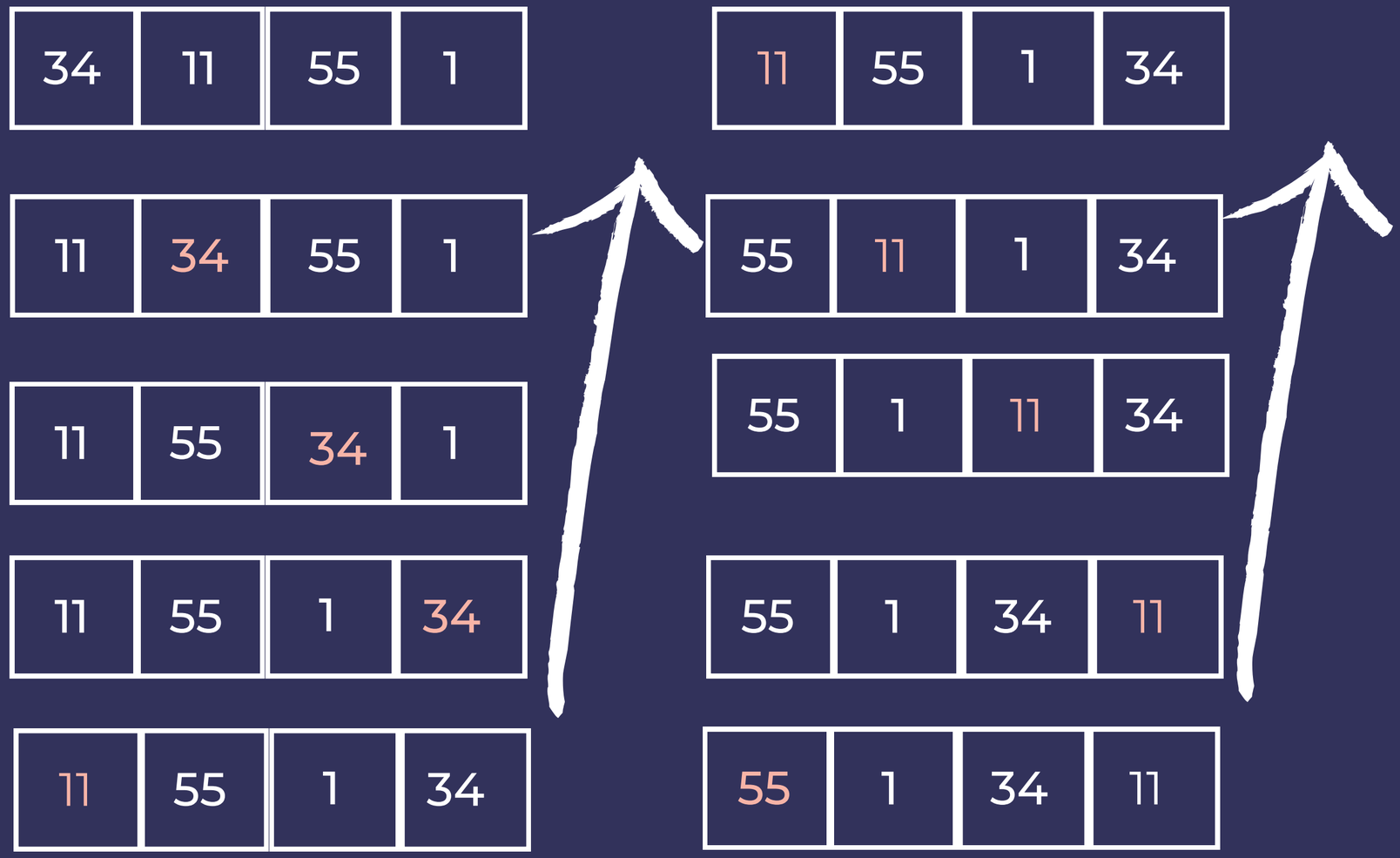
Bubble Sort è un algoritmo di ordinamento elementare che opera confrontando e scambiando ripetutamente coppie di elementi adiacenti se sono fuori ordine.

1. **Confronto:** L'algoritmo inizia con il primo elemento dell'array e lo confronta con il successivo. Se l'elemento successivo è più piccolo dell'elemento corrente, vengono scambiati.
2. **Iterazione:** I confronti e scambi avvengono per tutti gli elementi dell'array e continua a eseguire passate finché non se ne verifica una senza alcun scambio.
3. **Complessità temporale**: O(n^2)

Bacaro
Tech

CODE AND FUN

ESECUZIONE DEL BUBBLE SORT



Bacaro
Tech

CODE AND FUN

BUBBLE SORT PSEUDOCODICE

```
funzione bubbleSort(array):
    n = lunghezza dell'array
    fatto = vero
    finché fatto è vero:
        fatto = falso
        per i da 0 a n-2:
            se array[i] > array[i+1]:
                scambia array[i] e array[i+1]
                fatto = vero
```

Statistiche:

- tempo esecuzione medio O(n^2)
- tempo esecuzione nel caso peggiore O(n^2)
- spazio di occupazione O(1)

Bacaro
Tech

CODE AND FUN

QUICK SORT E IL DIVIDE ET IMPERA

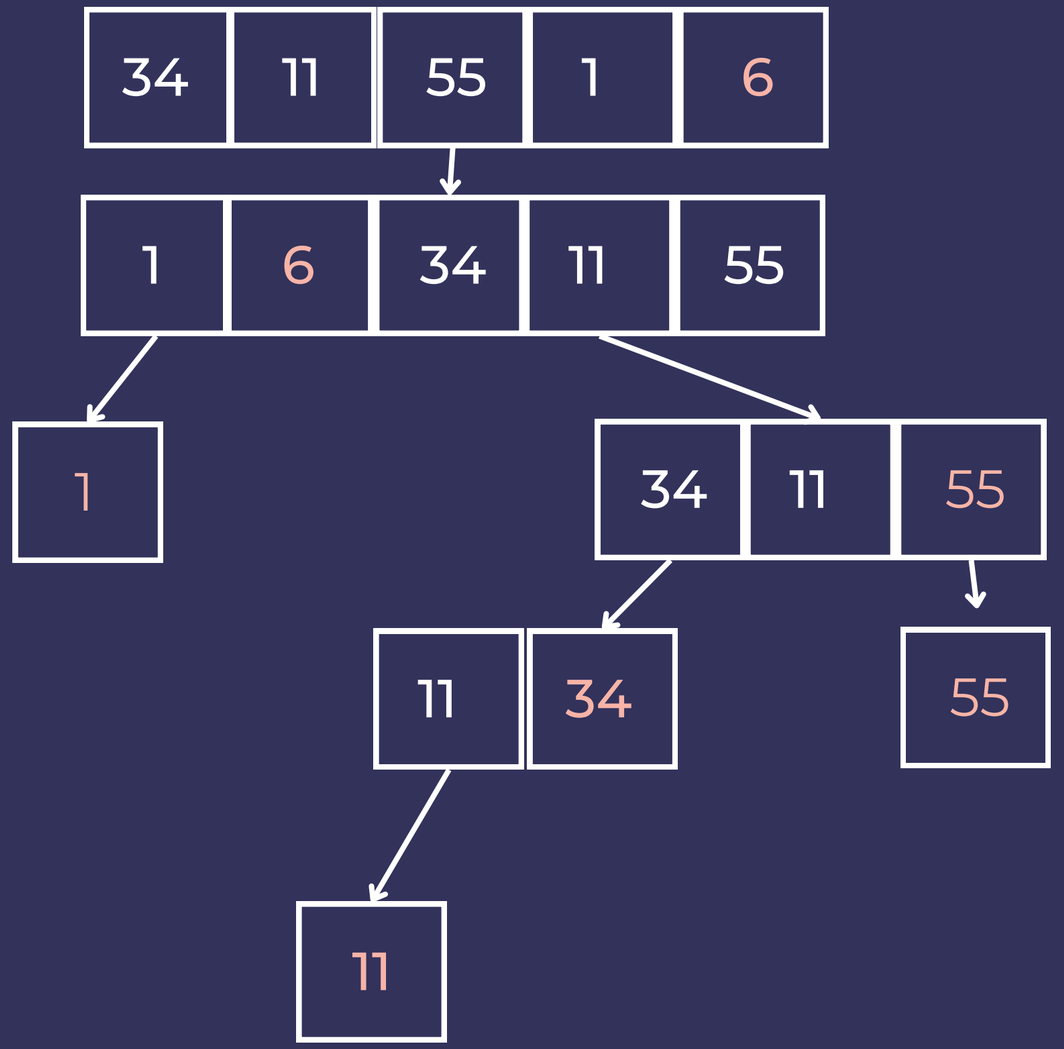
Il Quicksort è un algoritmo di ordinamento molto efficiente basato sul paradigma "divide et impera". Questo algoritmo è ampiamente utilizzato in pratica ed è noto per le sue prestazioni elevate.

1. **Confronto**: Il Quicksort funziona suddividendo una lista in due sotto-liste, quindi ordinando separatamente le due sotto-liste, a dx del pivot tutti gli elementi maggiori di esso e a sx del pivot tutti gli elementi minori di esso. Non per forza in ordine!
2. **Iterazione**: Questa suddivisione e ricorsione continua fino a quando tutte le sotto-liste sono di dimensione 0 o 1.
3. **Complessità temporale**: Il tempo medio è di O(logn)

Bacaro
Tech

CODE AND FUN

ESECUZIONE DEL QUICK SORT



Bacaro
Tech

CODE AND FUN

QUICK SORT PSEUDOCODICE

```
funzione quickSort(array):
    se la lunghezza dell'array è minore o uguale a 1
    allora restituisci l'array (è già ordinato)
    scegli un elemento come pivot (spesso è il primo elemento)
    crea tre liste vuote: meno, uguale, più
    per ogni elemento nell'array:
        se l'elemento è minore del pivot, aggiungilo a meno
        se l'elemento è uguale al pivot, aggiungilo a uguale
        se l'elemento è maggiore del pivot, aggiungilo a più
    restituisci concatenazione(quickSort(meno), uguale, quickSort(più))
```

Statistiche:

- tempo esecuzione medio O(nlogn)
- tempo esecuzione nel caso peggiore O(n^2)
- spazio di occupazione O(1)

Bacaro
Tech

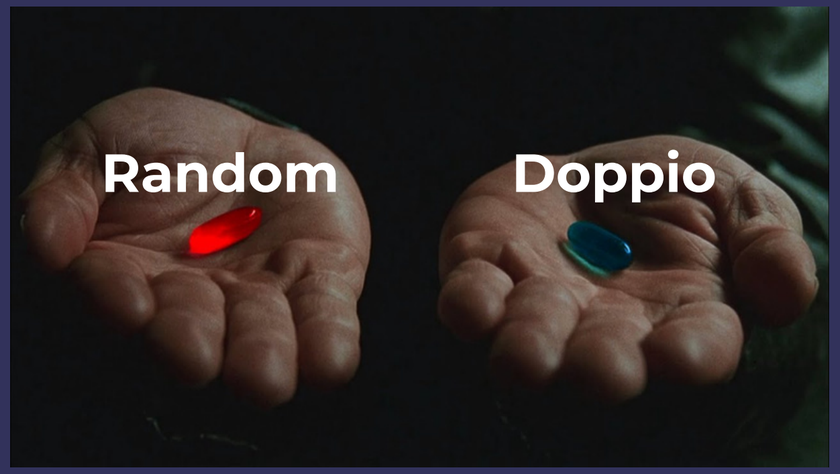
CODE AND FUN

QUALI SONO LE OTTIMIZZAZIONI DEL QUICK SORT

Il problema del quicksort e' che le prestazioni di quest'algoritmo variano in base alla scelta del pivot.

Esistono 2 tipologie di prestazioni:

- **Pivot randomico**: il pivot viene scelto random tra i vari elementi dell'array
- **Doppio pivot**: a ogni iterazione vengono scelti 2 pivot



Bacaro
Tech

CODE AND FUN

Vi ringraziamo per l’attenzione e per ogni domanda siamo qui per rispondervi!
Dopo di ché LEETCODATAAA e si gode!