

Titolo unità didattica: Strutture dati: array

[07]

Titolo modulo : Algoritmo per la determinazione
dell'uguaglianza di due array

[05-T]

Uguaglianza delle componenti di ugual posto di due array 1D

Argomenti trattati:

- ✓ definizione di uguaglianza di due array
- ✓ algoritmo incrementale per la determinazione dell'uguaglianza
- ✓ costo dell'algoritmo

Prerequisiti richiesti: AP-07-01-T

problema:

determinare se due array 1D sono **uguali**

due variabili **a** e **b** di tipo array (1D)
sono **uguali**

se hanno lo **stesso size** e

se **$a[i] = b[i]$** per tutti gli **i**

se per un **i** si ha che **$a[i]$** e **$b[i]$** sono diversi
allora i due array sono **diversi**;

altrimenti (cioè se per tutti gli **i**, **$a[i]$** e **$b[i]$**
sono uguali) i **due array sono uguali**

problema:

determinare se due array 1D sono **uguali**

dati di input: il primo array (variabile **a**), il secondo array (variabile **b**), il size degli array (variabile **n**)

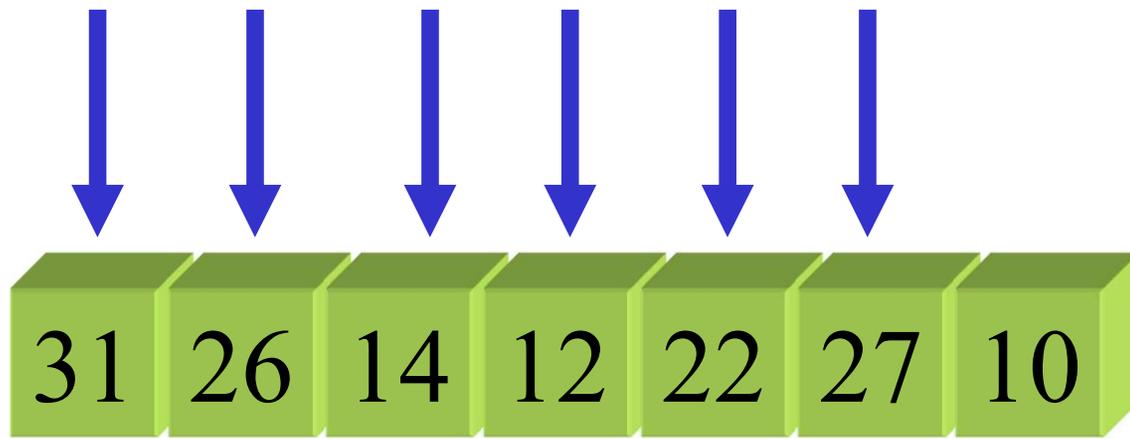
dato di output: **true** (uguali), **false** (diversi)
(variabile **uguale**)

costrutto ripetitivo: **do-while**

operazione ripetuta (al generico passo **i**):

confrontare **a[i]** e **b[i]**

se sono diversi o se sono state esaminate tutte le componenti dei due array, terminare il ciclo



true
uguale

```
uguale = true ;  
do {  
    if (a[i] != b[i]) {  
        uguale = false; }  
    i = i+1;  
} while (uguale && i < n )
```

a



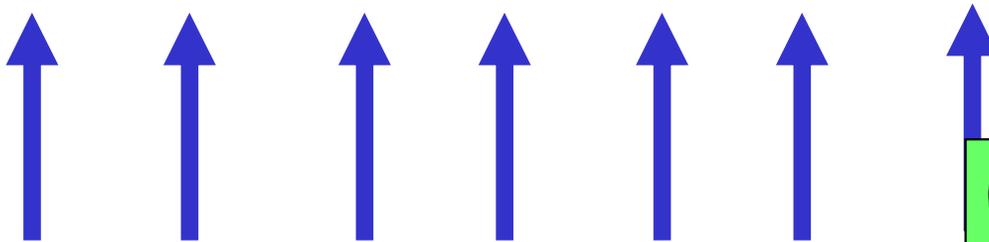
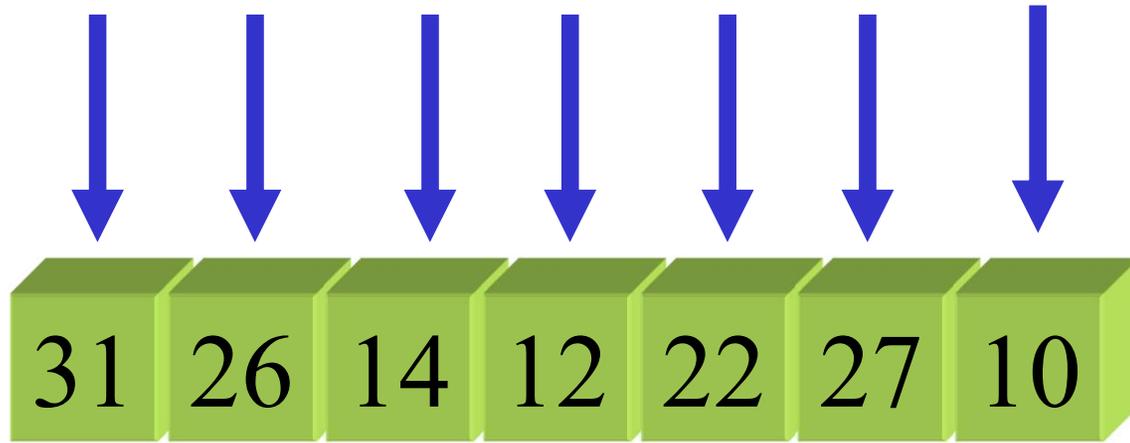
diversi

b



false
uguale

```
uguale = true ;  
do {  
    if (a[i] != b[i]) {  
        uguale = false ;  
    }  
    i = i+1 ;  
} while (uguale && i < n )
```



true

uguale

```
uguale = true ;  
do {  
    if (a[i] != b[i]) {  
        uguale = false; }  
    i = i+1;  
} while (uguale && i < n )
```

a



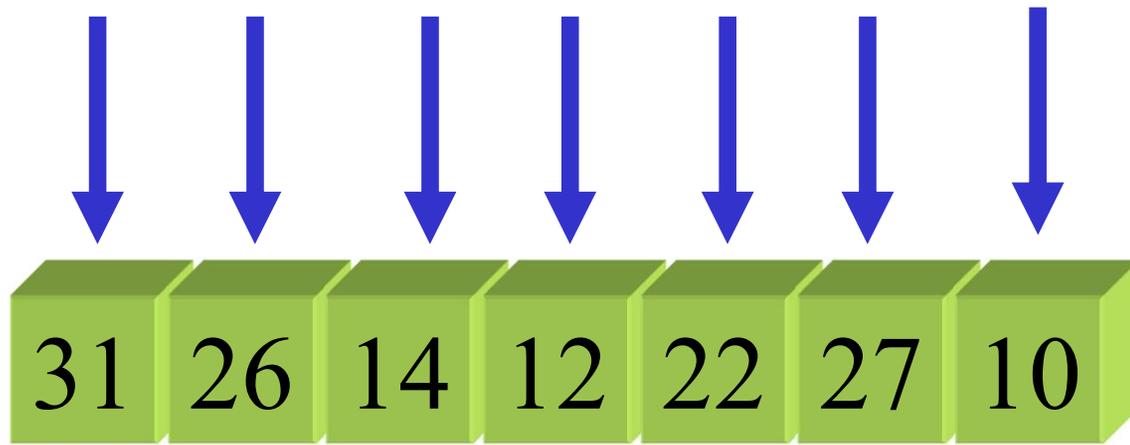
b



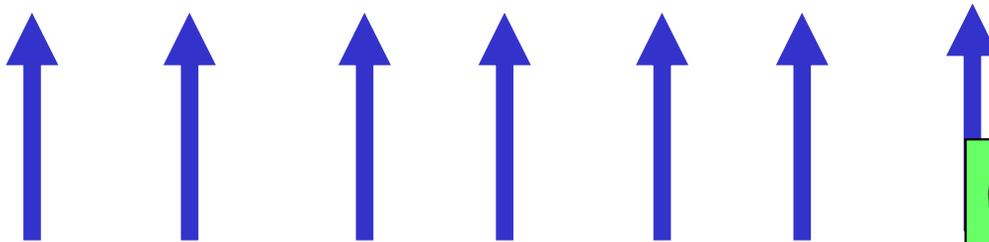
diversi

false
uguale

```
uguale = true ;  
do {  
    if (a[i] != b[i]) {  
        uguale = false; }  
    i = i+1;  
} while (uguale && i < n )
```



uguale



true
uguale

```
uguale = true ;  
do {  
    if (a[i] != b[i]) {  
        uguale = false ;  
    }  
    i = i+1 ;  
} while (uguale && i < n )
```

```
logical uguaglianza_array(char a[], char b[], int n) {  
    int i;  
    logical uguale;  
    i = 0 ;  
    uguale = true;  
    do {  
        if (a[i] != b[i])  
            { uguale = false ;}  
        i = i+1 ;  
    } while ( uguale && i < n) ;  
    return uguale ;  
}
```

versione 1

n
confronti tra gli elementi dei
due array
(al più)

uguale==true

```
logical uguaglianza_array(char a[], char b[], int n) {  
    int i;  
    logical uguale;  
    i = 0 ;  
    uguale = true;  
    while (uguale && i < n) {  
        if (a[i] != b[i])  
            { uguale = false;}  
        i = i+1;  
    }  
    return uguale;  
end
```

versione 2

n
confronti tra gli
elementi dei due
array
(al più)