

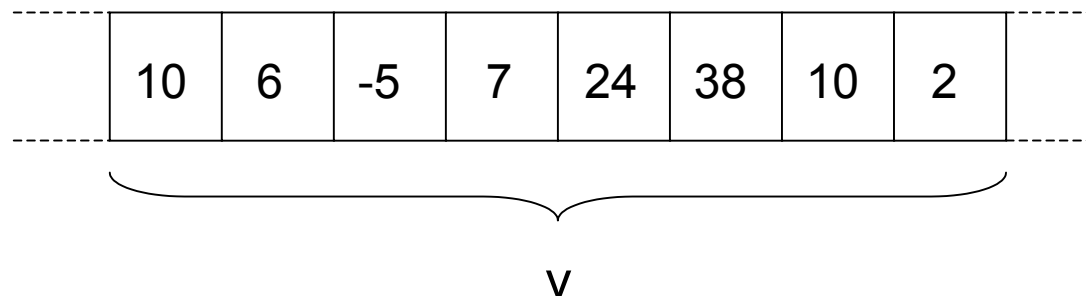
Array

In alcuni casi, l'informazione che bisogna elaborare consiste di un'aggregazione di valori, piuttosto che di un valore solo.

Questo significa che sarebbe conveniente indicare l'insieme di valori di interesse con una sola variabile piuttosto che con tante variabili quante sono i valori da considerare: una variabile di *tipo strutturato*.

In Fortran, questa possibilità è offerta dagli **array**.

Un array è un insieme di variabili, tutte dello stesso tipo, identificato da un nome unico. Gli elementi dell'array sono disposti in memoria in posizioni consecutive.



Per definire un array, è necessario specificare:

- il nome della variabile array
- il tipo cui gli elementi appartengono
- il numero degli elementi presenti (cardinalità dell'array)

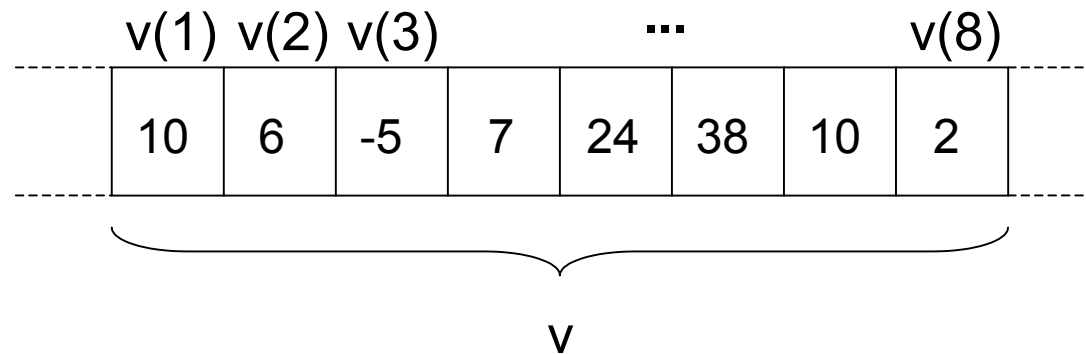
```
! un array di 50 elementi  
INTEGER :: v  
DIMENSION v(50)
```

```
! un array di num elementi  
INTEGER :: num  
REAL :: w  
PARAMETER (num=100)  
DIMENSION w(num)
```

costanti



Per accedere ai singoli elementi di un array, è necessario specificare il nome della variabile array e la posizione dell'elemento di interesse tramite un valore intero (variabile o costante) che si definisce *indice*.



Ogni elemento di un array è, a tutti gli effetti, una variabile del tipo costituente l'array.

$v(2) = 3$

WRITE (*, *) $v(7)$

$i = 3$

$a = v(i) * 4 + 6$

Inizializzazione di un array

```
Program InizArray
  IMPLICIT NONE

  INTEGER :: v,i,num
  PARAMETER (num=100)
  DIMENSION v (num)

  ! Inizializzazione dell'array v
  DO i=1,num
    v (i)=0
  END DO

  END PROGRAM
```

Cardinalità e riempimento

A volte non è possibile conoscere con precisione la dimensione di un array all'atto della scrittura del programma.

In questo caso, si fissa la cardinalità dell'array ad un valore ragionevole, che si suppone sia sufficiente per le esigenze del programma, e si adopera una variabile intera che conterrà, all'esecuzione, il numero di elementi effettivamente presenti nell'array.

```
Program InizArray
  IMPLICIT NONE
  INTEGER :: v,i,num,riemp
  PARAMETER(num=100)
  DIMENSION v(num)

  ! Inizializzazione dell'array v
  WRITE(*,*) "Quanti elementi ?"
  READ(*,*) riemp

  DO i=1,riemp
    WRITE(*,*) "Valore v(",i,"): "
    READ(*,*) v(i)
  END DO

END PROGRAM
```

Algoritmi definiti sugli array

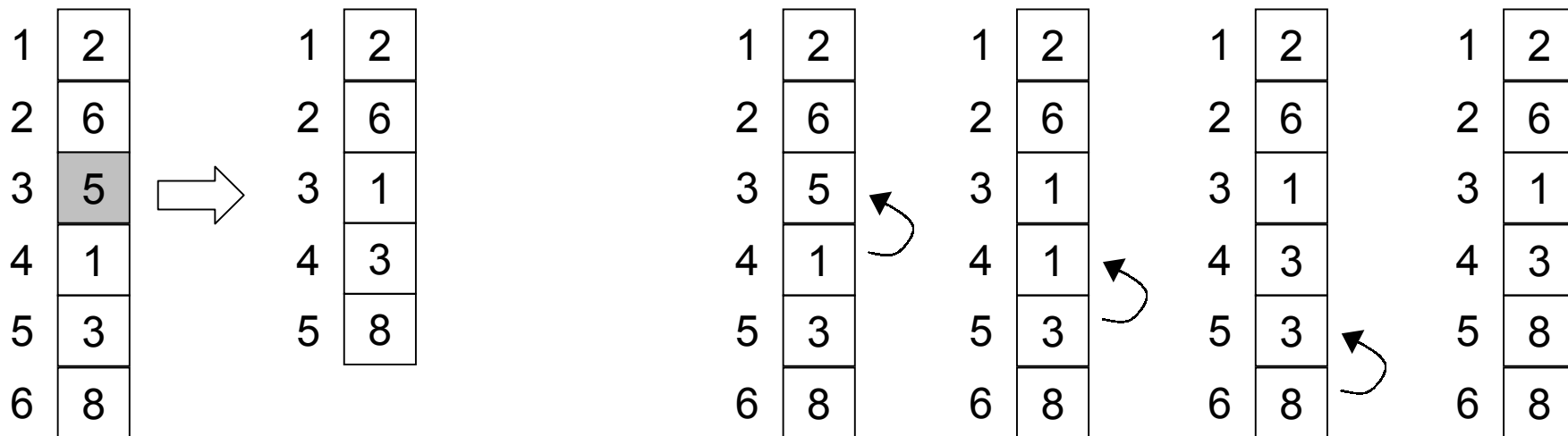
Per utilizzare in maniera efficiente gli array, è necessario poter eseguire alcune operazioni “tipiche”:

- inizializzazione
- lettura
- stampa
- ricerca del minimo e del massimo
- ricerca di un valore
- eliminazione di un valore
- inserimento di un valore
- ordinamento del vettore

Eliminazione di un valore da un array

Nel caso si debba eliminare un certo valore da un array, i passi da compiere sono:

- ricerca nell'array per identificare la posizione del valore (se presente)
- cancellazione logica tramite scorrimento di una posizione verso l'alto dei successivi elementi dell'array

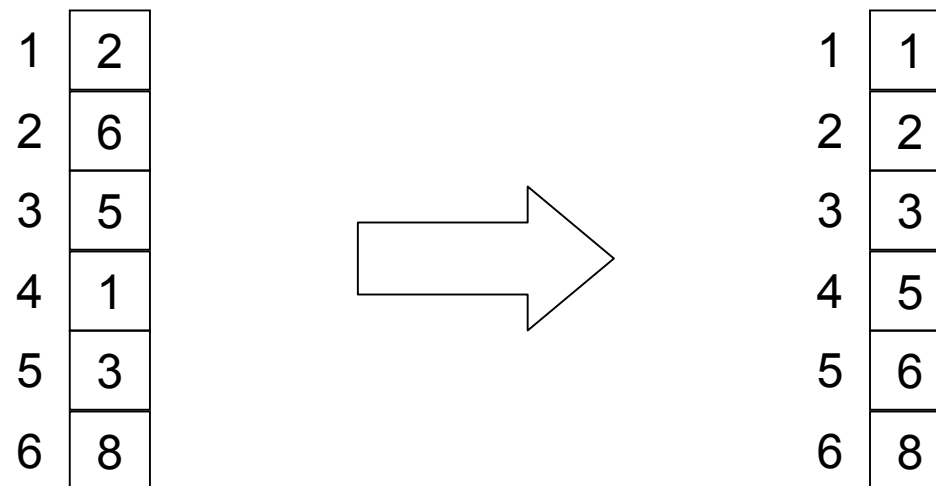


Ordinamento di un array

L'operazione di ordinamento consiste in una permutazione degli elementi nell'array in modo che, al termine dell'ordinamento, la disposizione degli elementi nell'array rispetti un ordine specifico (p.es. crescente).

Lo scopo dell'ordinamento è di facilitare successive ricerche di elementi nell'array che è stato ordinato.

Sono stati proposti numerosi algoritmi di ordinamento, con caratteristiche diverse. Non esiste l'algoritmo di ordinamento ottimo.



Ordinamento per selezione (select sort)

Questo algoritmo si basa sul seguente principio:

dato un array *vet* di *N* elementi, si determina l'elemento minimo tra *vet*(1), *vet*(2), ... , *vet*(*N*) e lo si scambia con il primo elemento

Queste operazioni vengono poi ripetute su *N*-1 elementi a partire da *vet*(2), poi su *N*-2 elementi a partire da *vet*(3), ..., su 2 elementi a partire da *vet*(*N*-1).

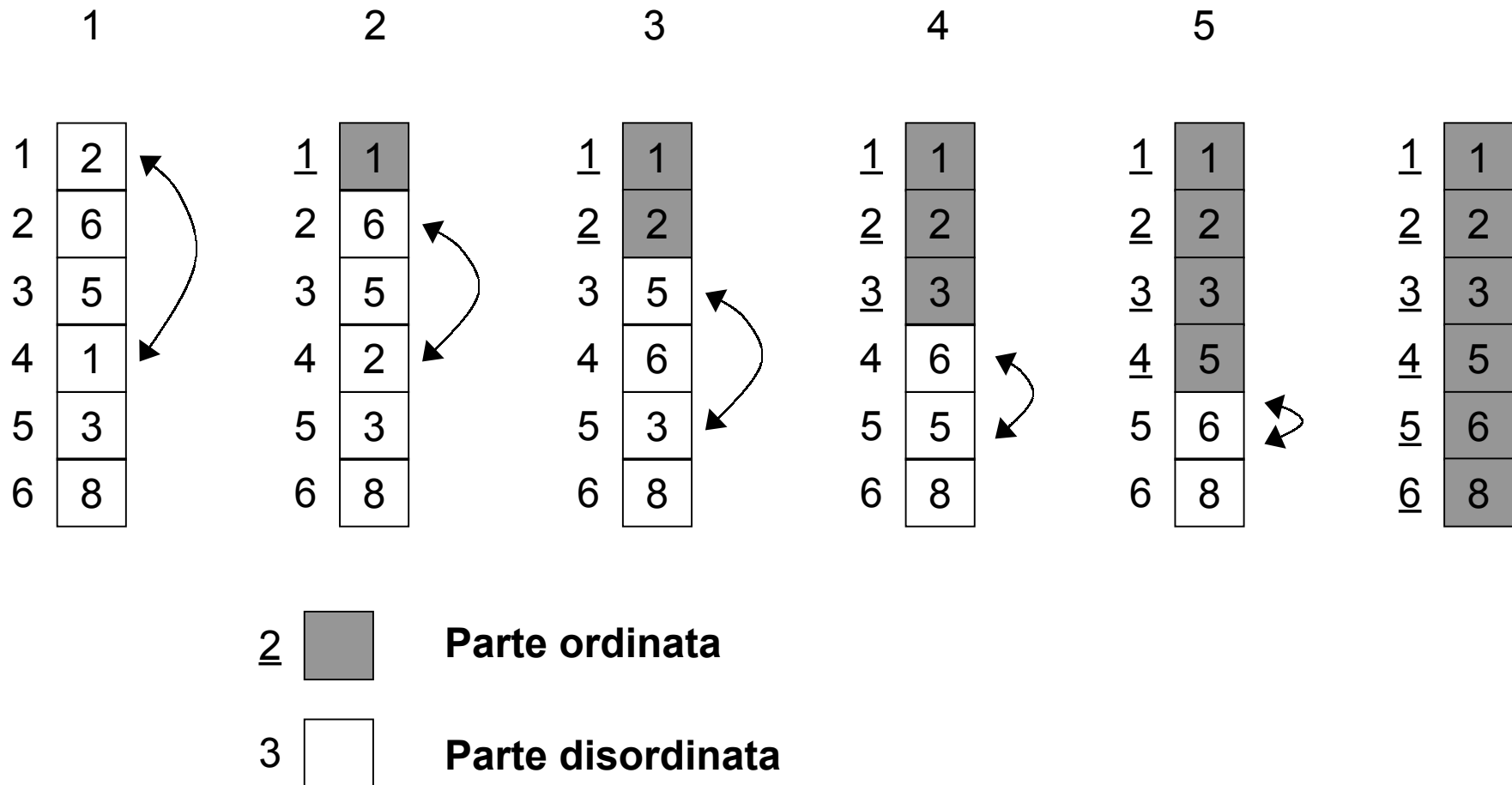
```
DO  i=1,N-1
```

```
    determina l'elemento minimo in vet(i),...,vet(N)  
    e scrivi in k la sua posizione
```

```
    scambia vet(i) e vet(k)
```

```
END DO
```

Ordinamento per selezione (select sort)



Inserimento di un valore in un array ordinato

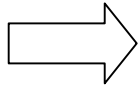
L'inserimento di un valore in un array ordinato deve mantenere l'ordine degli elementi. Per questo motivo, non è possibile il semplice accodamento del nuovo valore a quelli già presenti, ma è necessario modificare la disposizione degli elementi già presenti per fare in modo che il nuovo valore possa essere inserito nella giusta posizione.

L'inserimento si realizza in due passi:

- individuare la posizione che deve assumere il nuovo valore all'interno dell'array per mantenere l'ordine dell'insieme dei valori
- rendere disponibile quella posizione facendo scorrere tutti gli elementi da quella posizione in poi di un posto in basso e infine scrivere il nuovo valore nella posizione resa disponibile

Inserimento di un valore in un array ordinato

1	1
2	2
3	3
4	5
5	6
6	8



1	1
2	2
3	3
4	4
5	5
6	6
7	8

1	1
2	2
3	3
4	5
5	6
6	8
7	

1	1
2	2
3	3
4	5
5	6
6	8
7	8

1	1
2	2
3	3
4	5
5	6
6	6
7	8

1	1
2	2
3	3
4	5
5	5
6	6
7	8

← 4