

Algoritmo per la ricerca di un elemento in un vettore

1. Introduzione e requisiti del problema
2. Specifica
3. Progetto della soluzione
4. Codifica

1. Introduzione e requisiti del problema

In questa esercitazione vengono presentati alcuni *Algoritmi* per la ricerca di un *elemento all'interno di un vettore*.

Spesso, infatti, è necessario cercare dei dati con determinate proprietà all'interno di una collezione, che può contenere i dati in disordine oppure in ordine (crescente o decrescente).

Nell'esercitazione sull'ordinamento (n.13) sono presentati diversi algoritmi per ordinare un insieme di dati non ordinato.

1. Introduzione e requisiti del problema

Esistono diversi algoritmi di ricerca di un elemento in un vettore. I principali sono i seguenti:

1. algoritmo di ricerca in un vettore non ordinato
2. algoritmo di ricerca sequenziale in un vettore ordinato
3. algoritmo di ricerca binaria in un vettore ordinato.

1. Introduzione e requisiti del problema

Requisiti del problema

Scrivere un programma che contenga le seguenti funzioni:

1. una funzione **leggiVettore**, per introdurre da tastiera il contenuto di un vettore
2. una funzione **stampaVettore**, per stamparne il contenuto
3. una funzione **ricerca** per ricercare un elemento di un vettore, presentata in tre versioni corrispondenti ai tre tipi di algoritmi di ricerca sopra citati.

1. Introduzione e requisiti del problema

Distinguiamo i tre casi di:

1. algoritmo di ricerca in un vettore non ordinato
2. algoritmo di ricerca sequenziale in un vettore ordinato
3. algoritmo di ricerca binaria in un vettore ordinato

1. Introduzione e requisiti del problema

1. Ricerca di un elemento in un vettore non ordinato

Esempio 1

Vettore non ordinato=

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

Ricerca dell'elemento= 7

Posizione(indice) in cui si trova l'elemento
cercato= 4

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

Esempio1

Vettore non ordinato=

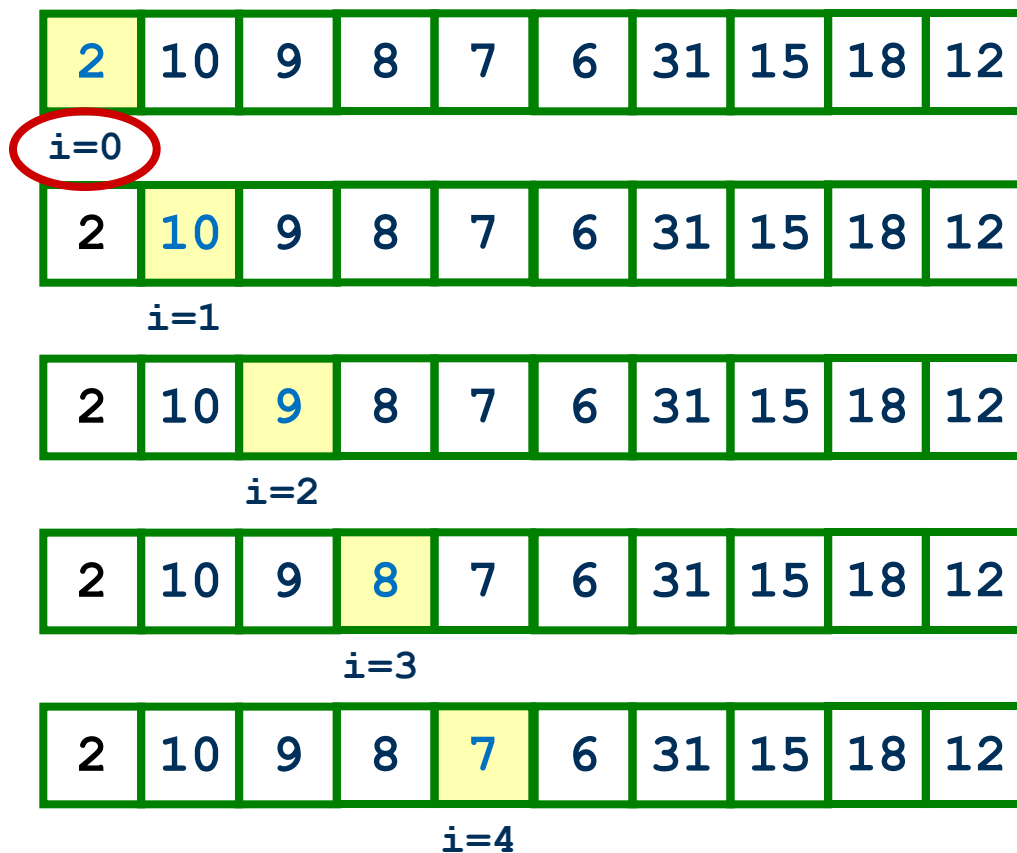
2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

Ricerca dell'elemento= 7

Posizione(indice) in cui si trova l'elemento
cercato= 4

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato



2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato



2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=0$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=1$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=2$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=3$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=4$

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=0$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=1$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=2$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=3$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=4$

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=0$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=1$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=2$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=3$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=4$

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=0$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=1$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=2$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=3$

2	10	9	8	7	6	31	15	18	12
---	----	---	---	---	---	----	----	----	----

 $i=4$

*il valore coincide con
quello cercato*

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

Esempio 2

Vettore non ordinato=

2	10	9	8	7	6	31
---	----	---	---	---	---	----

Ricerca dell'elemento= 5

Posizione(indice) in cui si trova l'elemento
cercato= non trovato

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=0

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=1

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=2

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=3

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=4

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=5

2	10	9	8	7	6	31
---	----	---	---	---	---	----

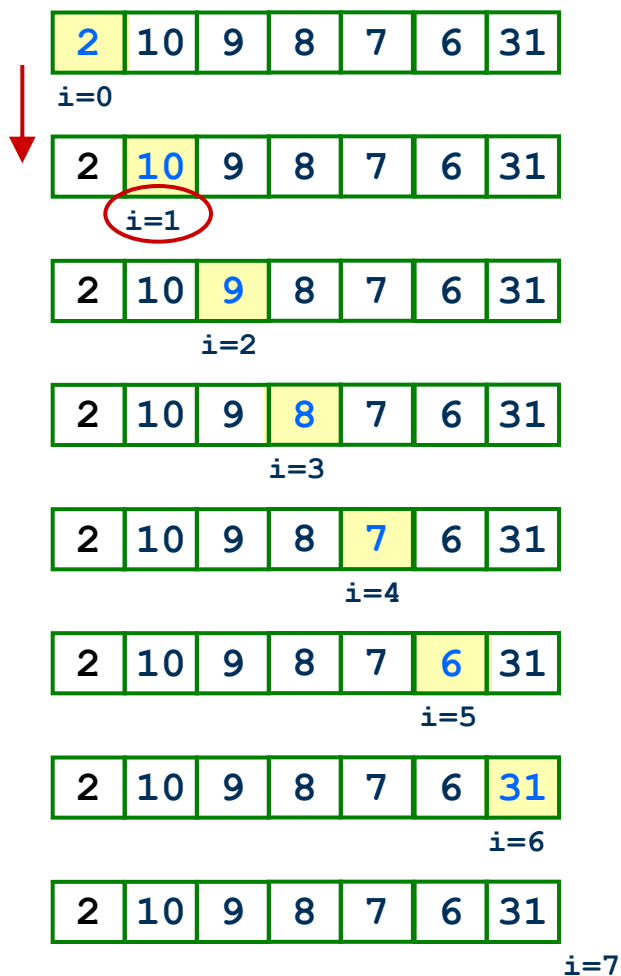
i=6

2	10	9	8	7	6	31
---	----	---	---	---	---	----

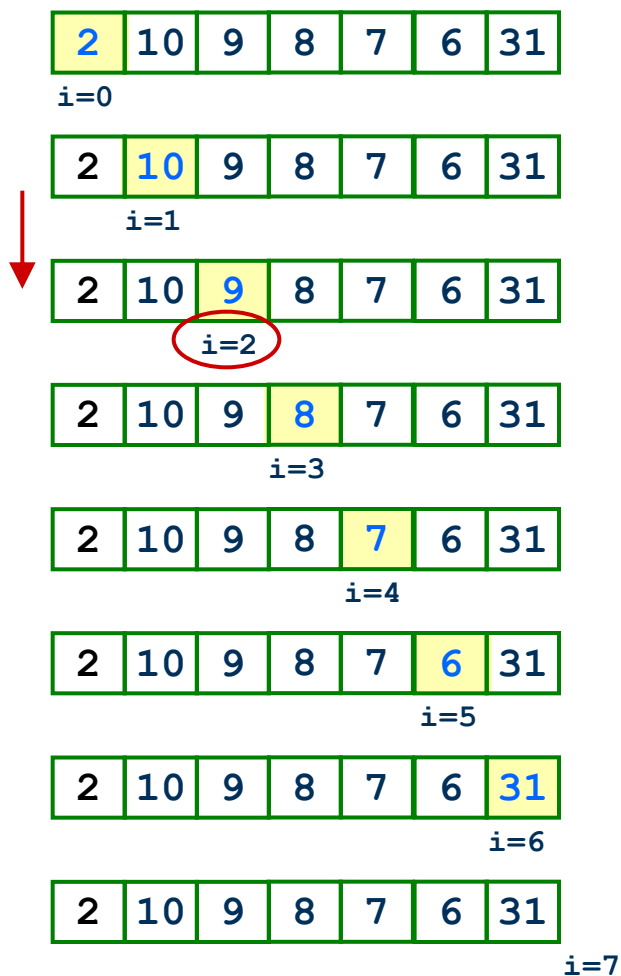
i=7

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

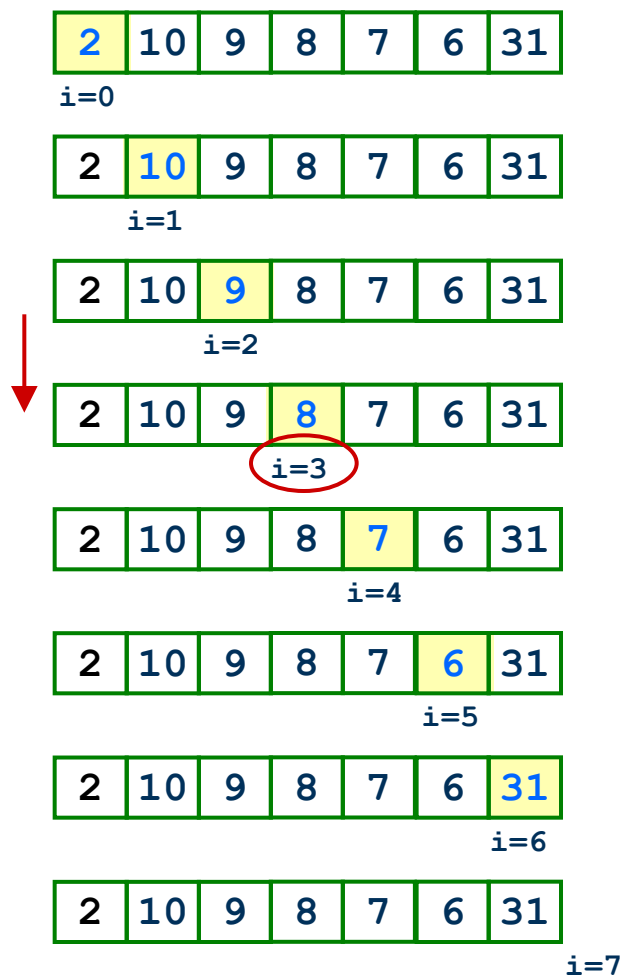


1. Ricerca di un elemento in un vettore non ordinato



2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato



2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=0$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=1$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=2$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=3$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=4$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=5$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=6$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=7$

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=0$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=1$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=2$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=3$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=4$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=5$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=6$

2	10	9	8	7	6	31
---	----	---	---	---	---	----

 $i=7$

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=0

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=1

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=2

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=3

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=4

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=5



2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=6

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=7

*L'uguaglianza non
è verificata*

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=0

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=1

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=2

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=3

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=4

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=5

2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=6



2	10	9	8	7	6	31
---	----	---	---	---	---	----

i=7

*L'uguaglianza non
è verificata*

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

La ricerca di un elemento **elem** in un vettore **vett** non ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non termina l'array
2. Non si trova l'elemento cercato

```
i=0;  
while ((i<DIM) && (vett[i]!=elem))  
    i++;
```

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

La ricerca di un elemento **elem** in un vettore **vett** non ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non termina l'array
2. Non si trova l'elemento cercato

```
i=0;  
while ((i<DIM) && (vett[i]!=elem))  
    i++;
```


2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

La ricerca di un elemento **elem** in un vettore **vett** non ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non termina l'array
2. Non si trova l'elemento cercato

```
i=0;  
while ((i<DIM) && (vett[i]!=elem))  
    i++;
```

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

La ricerca di un elemento **elem** in un vettore **vett** non ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non termina l'array
2. Non si trova l'elemento cercato

```
i=0;  
while ((i<DIM) && (vett[i]!=elem))  
    i++;
```

2. Progetto della soluzione

1. Ricerca di un elemento in un vettore non ordinato

La ricerca di un elemento **elem** in un vettore **vett** non ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non termina l'array
2. Non si trova l'elemento cercato

```
i=0;  
while ((i<DIM) && (vett[i]!=elem))  
    i++;
```

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

Esempio 1

Vettore ordinato=

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

Ricerca dell'elemento=11

Posizione(indice) in cui si trova l'elemento
cercato= non trovato

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=0

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=1

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=2

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=3

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=4

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=5

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=6

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato



2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato



2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=0$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=1$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=2$ 

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=3$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=4$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=5$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=6$

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=0$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=1$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=2$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=3$



2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=4$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=5$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=6$

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=0$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=1$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=2$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=3$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=4$ 

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=5$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

 $i=6$

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=0$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=1$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=2$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=3$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=4$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=5$

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

$i=6$

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=0

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=1

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=2

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=3

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=4

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=5

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=6

*l'elemento cercato non
è presente nell'array*

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=0

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=1

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=2

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=3

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=4

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=5

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

i=6

i=10

Esempio:*si cerchi il numero 35*

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

La ricerca di un elemento **elem** in un vettore **vett** ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non si trova l'elemento cercato
2. Non si trova un elemento maggiore di quello cercato
3. Non termina l'array

```
i=0;  
while ((i<DIM) && (vett[i]<elem))  
    i++;
```

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

La ricerca di un elemento **elem** in un vettore **vett** ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non si trova l'elemento cercato
2. Non si trova un elemento maggiore di quello cercato
3. Non termina l'array

```
i=0;  
while ((i<DIM) && (vett[i]<elem))  
    i++;
```

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

La ricerca di un elemento **elem** in un vettore **vett** ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non si trova l'elemento cercato
2. Non si trova un elemento maggiore di quello cercato
3. Non termina l'array

```
i=0;  
while ((i<DIM) && (vett[i]<elem))  
    i++;
```


2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

La ricerca di un elemento **elem** in un vettore **vett** ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non si trova l'elemento cercato
2. Non si trova un elemento maggiore di quello cercato
3. Non termina l'array

```
i=0;  
while ((i<DIM) && (vett[i]<elem))  
    i++;
```

2. Progetto della soluzione

2. Ricerca di un elemento in un vettore ordinato

La ricerca di un elemento **elem** in un vettore **vett** ordinato contenente DIM elementi richiede la visita in sequenza degli elementi dell'array finché:

1. Non si trova l'elemento cercato
2. Non si trova un elemento maggiore di quello cercato
3. Non termina l'array

```
i=0;  
while ((i<DIM) && (vett[i]<elem))  
    i++;
```

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

Esempio 1

Vettore ordinato=

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

Ricerca dell'elemento= 7

Posizione(indice) in cui si trova l'elemento
cercato= 2

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

Esempio 1

Vettore ordinato=

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

Ricerca dell'elemento= 7

Posizione(indice) in cui si trova l'elemento
cercato= 2



2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

I=0

S=9

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

I=0

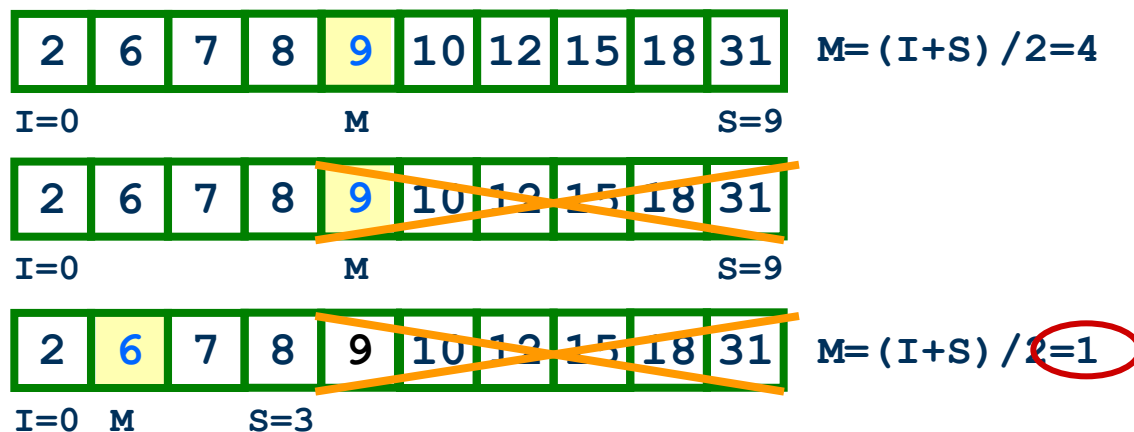
M

S=9

$$M = (I + S) / 2 = 4$$

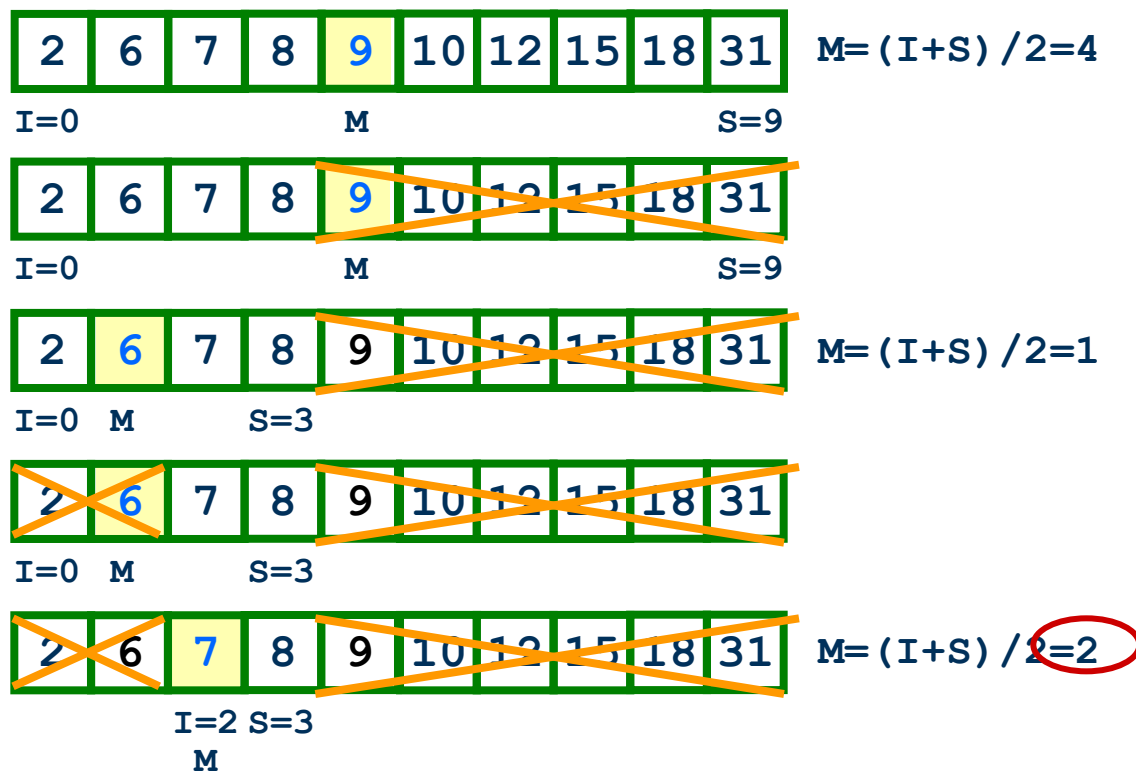
2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato



2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato



L'elemento è stato trovato in posizione M

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

Esempio 2

Vettore ordinato=

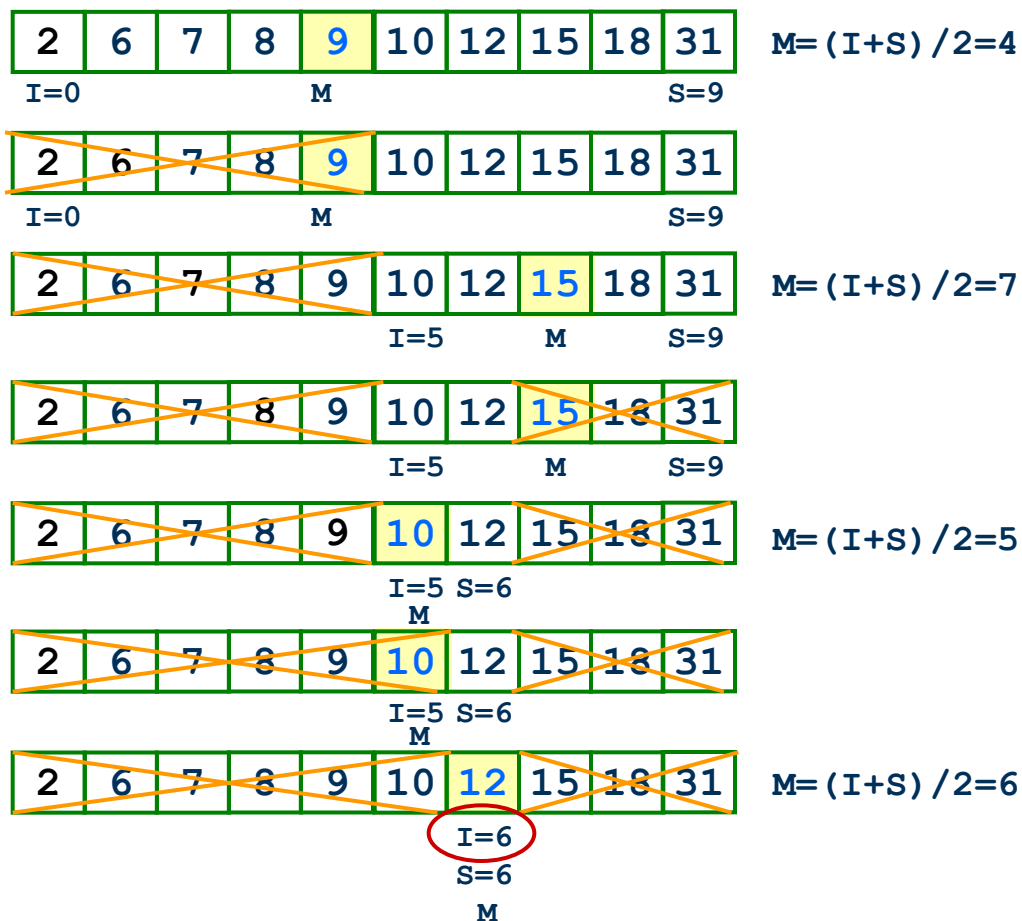
2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

Ricerca dell'elemento=11

Posizione(indice) in cui si trova l'elemento
cercato= non trovato

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato



2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

I=6
S=6
M

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

S=5 I=6

se cerco il valore 11

2. Progetto della soluzione

3. Ricerca binaria di un elemento in un vettore ordinato

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

I=6
S=6
M

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

S=5 I=6

se cerco il valore 11

2	6	7	8	9	10	12	15	18	31
---	---	---	---	---	----	----	----	----	----

S=6 I=7

se cerco il valore 14

2. Progetto della soluzione

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

2. Progetto della soluzione

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

```
while (I<=S)
{
    M=(I+S)/2
    if (valore in posizione M > valore da cercare)
        Scarta la metà iniziale dell'array: I=M+1
    else if (valore in posizione M < valore da cercare)
        Scarta la seconda metà dell'array: S=M-1
    else l'elemento cercato si trova in posizione M
}
```

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

2. Progetto della soluzione

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

2. Progetto della soluzione

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

 if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

 else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

 else l'elemento cercato si trova in posizione M

}

Pseudocodice

Porzione dell'array da considerare:

I=primo indice dell'array

S=ultima posizione dell'array

while (I<=S)

{

$M = (I + S) / 2$

if (valore in posizione M > valore da cercare)

 Scarta la metà iniziale dell'array: $I = M + 1$

else if (valore in posizione M < valore da cercare)

 Scarta la seconda metà dell'array: $S = M - 1$

else l'elemento cercato si trova in posizione M

}

2. Progetto della soluzione

Nota sui tempi di esecuzione degli algoritmi

Gli algoritmi di *ricerca sequenziale* non sono molto efficienti: il tempo di esecuzione è proporzionale alla lunghezza della sequenza della collezione, ed in particolare varia in modo lineare. Ovvero, se si raddoppia la lunghezza della sequenza, si raddoppia il tempo medio di esecuzione.

Se la lunghezza aumenta 10 volte, aumenta 10 volte anche il tempo di esecuzione.

Se n è la lunghezza della sequenza l'algoritmo richiede in media $n/2$ confronti tra l'elemento da cercare e gli elementi della collezione (considerando ugualmente probabili tutti gli *elementi*).

(continua)

2. Progetto della soluzione

Nota sui tempi di esecuzione degli algoritmi

Il tempo di esecuzione dell'algoritmo di *ricerca binaria* aumenta in modo meno che lineare all'aumentare della lunghezza della sequenza della collezione.

Ad ogni iterazione dell'algoritmo si riescono infatti ad escludere dalla ricerca un numero di elementi pari alla metà della parte di collezione che si sta analizzando.

Se n è la lunghezza della sequenza l'algoritmo richiede al più $\log_2 n$ confronti.