

ESERCIZIO: Ricerca di un elemento

Sapendo che il vettore è **ordinato**, la ricerca può essere ottimizzata.

– **Vettore ordinato in senso non decrescente:**

Esiste una relazione d'ordine totale sul dominio degli elementi

se $i < j$ si ha $V[i] \leq V[j]$

– **Vettore ordinato in senso crescente:**

se $i < j$ si ha $V[i] < V[j]$

In modo analogo si definiscono l'ordinamento in senso **non crescente** e **decrescente**

ESERCIZIO: RICERCA BINARIA

Ricerca binaria di un elemento in un vettore ordinato in senso non decrescente in cui il primo elemento è **first** e l'ultimo **last**

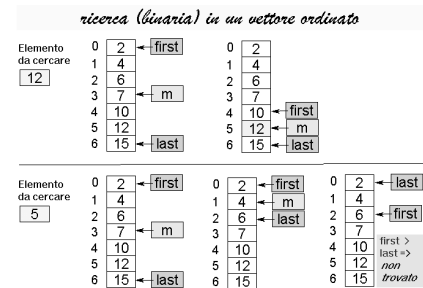
La tecnica di **ricerca binaria**, rispetto alla ricerca esaustiva, consente di **eliminare ad ogni passo metà degli elementi del vettore**

ESERCIZIO: RICERCA BINARIA

- Si confronta l'elemento cercato **e1** con quello mediano del vettore, **v[med]**
- Se **e1 == v[med]**, fine della ricerca (**trovato=true**)
- Altrimenti, se il vettore ha almeno due componenti (**first < last**):
 - se **e1 < v[med]**, ripeti la ricerca nella prima metà del vettore (indici da **first** a **med-1**)
 - se **e1 > v[med]**, ripeti la ricerca nella seconda metà del vettore (indici da **med+1** a **last**)

ESERCIZIO: RICERCA BINARIA

Esempio



ESERCIZIO: RICERCA BINARIA

```
int ricerca_bin (int vet[], int e1)
{int first=0, last=N-1, med=(first+last)/2;
 int T=0;
 while ((first<=last) && (T==0))
 { if (e1==vet[med])
    T=1;
   else
    if (e1 < vet[med]) last=med-1;
    else first=med+1;
    med = (first + last) / 2;
 }
 return T;
}
```

ESERCIZIO: Ricerca binaria di un elemento

```
#include <stdio.h>
#define N 15

int ricerca (int vet[], int e1);
main ()
{int i;
 int a[N];
 printf ("Scrivi %d numeri interi ordinati\n", N);
 for (i = 0; i<N; i++)
   scanf ("%d", &a[i]);
 printf ("Valore da cercare: ");
 scanf ("%d", &i);
 if (ricerca_bin(a,i)) printf("\nTrovato\n");
 else printf("\nNon trovato\n");
}
```

OSSERVAZIONI

Si noti che la ricerca binaria può essere definita facilmente in ***modo ricorsivo***

Si noti infatti che si effettua un confronto dell'elemento cercato *el* con l'elemento medio del vettore *V[med]*

- Se l'elemento cercato è uguale si termina (caso base)
- Altrimenti se *el < V[med]* si effettua una ricerca binaria sulla prima metà del vettore
- Altrimenti (se *el > V[med]*) si effettua una ricerca binaria sulla seconda metà del vettore

Esercizio: si scriva una procedura di ***ricerca binaria ricorsiva***

OSSERVAZIONI

Come generalizzare al caso di un array di elementi generici (non solo int)?

Provare ad adattare l'algoritmo

- usare il tipo `element`
- eliminare dalle funzioni `ricerca_bin` ogni dipendenza dal tipo `int` (operatori relazionali di confronto, ...), inserendo chiamate a funzioni relazionali appropriate (`isEqual`, `isLess`, ...)