

## La ricerca dicotomica

Se l'array è ordinato, la ricerca di un elemento può essere realizzata in modo più efficiente rispetto alla ricerca sequenziale, utilizzando la **ricerca dicotomica**. Si usa il termine *dicotomica* perché si procede a divisioni successive dell'array. La ricerca dicotomica presuppone che gli elementi del vettore siano ordinati.

Questo metodo di ricerca è un modo naturale di operare con insiemi ordinati. Per esempio, se vogliamo cercare nel dizionario il significato della parola *componente*, apriamo il dizionario nella parte iniziale, perché i termini del dizionario sono ordinati alfabeticamente, e il termine da ricercare inizia con la lettera C.

Se aprendo il dizionario all'inizio leggiamo per esempio la parola *edera*, continuiamo a cercare nella parte più a sinistra perché *edera* viene dopo *componente*. Se nella parte considerata leggiamo *banca*, continuiamo la ricerca nella parte più a destra, perché *banca* viene prima di *componente*. Ripetendo il procedimento su parti sempre più piccole e scartando le parti che non possono contenere il termine cercato, arriviamo a una pagina singola e possiamo stabilire se la parola *componente* è presente oppure non presente nel dizionario. Lo stesso metodo viene utilizzato per cercare un nome nell'elenco telefonico, oppure un nome in un elenco di persone ordinato alfabeticamente.

Formalizzando il procedimento, si divide l'insieme dei dati su cui ricercare esattamente a metà e si considera la parte che interessa. La parte rimasta viene ancora divisa a metà e così via.

Se scartiamo la parte a sinistra, la metà diventa il nuovo estremo sinistro della ricerca; se scartiamo la parte a destra, la metà diventa il nuovo estremo destro della ricerca. La divisione a metà dell'intervallo di ricerca continua finché l'elemento viene trovato e in questo caso si ottiene come risultato la posizione dell'elemento nell'array, oppure finché l'intervallo di ricerca si riduce a due elementi consecutivi dell'array e viene visualizzato un messaggio di "elemento non trovato".

In sostanza, dell'array in cui si vuole operare la ricerca si considerano soltanto gli estremi sinistro e destro e l'elemento che ha posizione centrale rispetto ai due estremi: se l'elemento da ricercare è uguale a uno dei tre elementi dell'array considerati, la ricerca è andata a buon fine.

In caso contrario si procede all'iterazione dei seguenti passi: se l'elemento da ricercare è maggiore dell'elemento di posizione centrale, questo diventa il nuovo estremo sinistro in cui operare la ricerca, altrimenti l'elemento di posizione centrale diventa il nuovo estremo destro; dopo aver adattato di conseguenza la posizione del nuovo elemento centrale, si riesegue il confronto precedente.

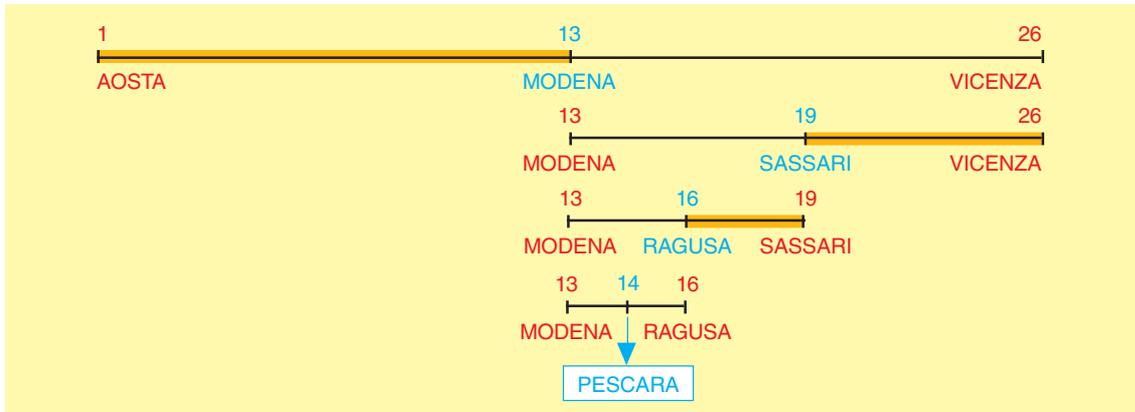
L'iterazione continua finché la ricerca ha avuto buon fine oppure le continue riduzioni dell'intervallo di ricerca fanno coincidere i due estremi.

Ad ogni passo dell'algoritmo i valori entro i quali effettuare la ricerca si dimezzano (all'incirca, in quanto i valori possono essere dispari). In breve le possibilità saranno una o due. In tal caso o il valore ricercato è tra i rimanenti, oppure non è presente nell'intero array.

Per esempio si supponga di dover cercare la città di PESCARA in un elenco alfabetico di città.

1. AOSTA	8. LECCE	15. PISA	22. TREVISO
2. BELLUNO	9. LIVORNO	16. RAGUSA	23. VARESE
3. CUNEO	10. LUCCA	17. ROVIGO	24. VENEZIA
4. ENNA	11. MASSA CARRARA	18. SALERNO	25. VERONA
5. FIRENZE	12. MILANO	19. SASSARI	26. VICENZA
6. FOGGIA	13. MODENA	20. SIRACUSA	
7. FORLI'	14. PESCARA	21. TARANTO	

I numeri si riferiscono alla posizione che la città occupa nell'elenco. Il segmento in colore rappresenta la parte dell'elenco che sicuramente non contiene la città cercata e che quindi non viene più presa in considerazione.



Si descrivono i vari passaggi dell'esempio attraverso la seguente tabella:

Step	Estremo sinistro	Estremo destro	Valore centrale	Esito	Operazione
1	AOSTA	VICENZA	MODENA	MODENA < PESCARA	Estremo Sinistro = MODENA
2	MODENA	VICENZA	SASSARI	SASSARI > PESCARA	Estremo Destro = SASSARI
3	MODENA	SASSARI	RAGUSA	RAGUSA > PESCARA	Estremo Destro = RAGUSA
4	MODENA	RAGUSA	PESCARA	PESCARA = PESCARA	FINE

## Progetto

### Ricerca un elemento in un array ordinato.

Il programma è in grado di ricercare un elemento all'interno di un vettore ordinato utilizzando l'algoritmo di ricerca dicotomica. Il vettore viene caricato utilizzando le stesse procedure del Progetto 1. Il programma termina visualizzando l'indice al quale si trova l'elemento ricercato oppure il messaggio "Il valore cercato non è presente".

#### Dati di input:

vettore di stringhe  
valore da ricercare

#### Dati di output:

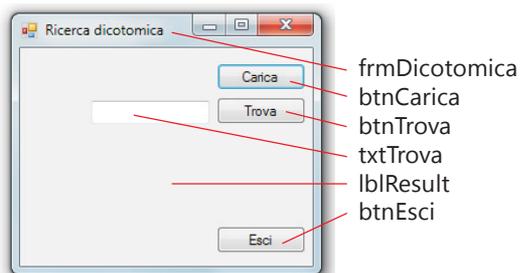
posizione dove si trova il valore cercato  
oppure  
messaggio di non trovato.

#### Nome del progetto

RicercaSequenziale di tipo Applicazione Windows Form

#### Disegno dell'interfaccia grafica

Classe	Proprietà dell'oggetto	
Form	Name	frmDicotomica
	Text	Ricerca dicotomica
TextBox	Name	txtTrova
Label	Name	lblResult
	Text	
Button	Name	btnCarica
	Text	Carica
Button	Name	btnTrova
	Text	Trova
Button	Name	btnEsci
	Text	Esci



### Gestione degli eventi

Il vettore viene caricato come nei progetti precedenti. Dopo aver caricato il vettore, viene inserito il testo nella casella di ricerca e si fa clic sul pulsante *Trova* per avviare la procedura di ricerca.

### Algoritmo in pseudocodifica

L'algoritmo verifica se è presente il termine da ricercare. Inizia la ricerca utilizzando una ripetizione postcondizionale.

La ripetizione termina quando la variabile booleana *trovato* assume il valore vero oppure quando l'estremo destro della ricerca diventa minore o uguale all'estremo sinistro.

In caso negativo avvisa l'utente.

inizio

    immetti cercato

    trovato = falso

    sx = 0

    dx = dimensione - 1

    esegui

        md = ( dx + sx ) \ 2

        se vettore(sx) = cercato O vettore(md) = cercato O vettore(dx) = cercato

        allora

            trovato = vero

        altrimenti

            se vettore(md) < cercato

                allora

                    sx = md + 1

                altrimenti

                    dx = md - 1

                fine se

        fine se

    ripeti finchè trovato = vero O dx <= sx

    se trovato = vero

    allora

        scrivi "il valore è stato trovato"

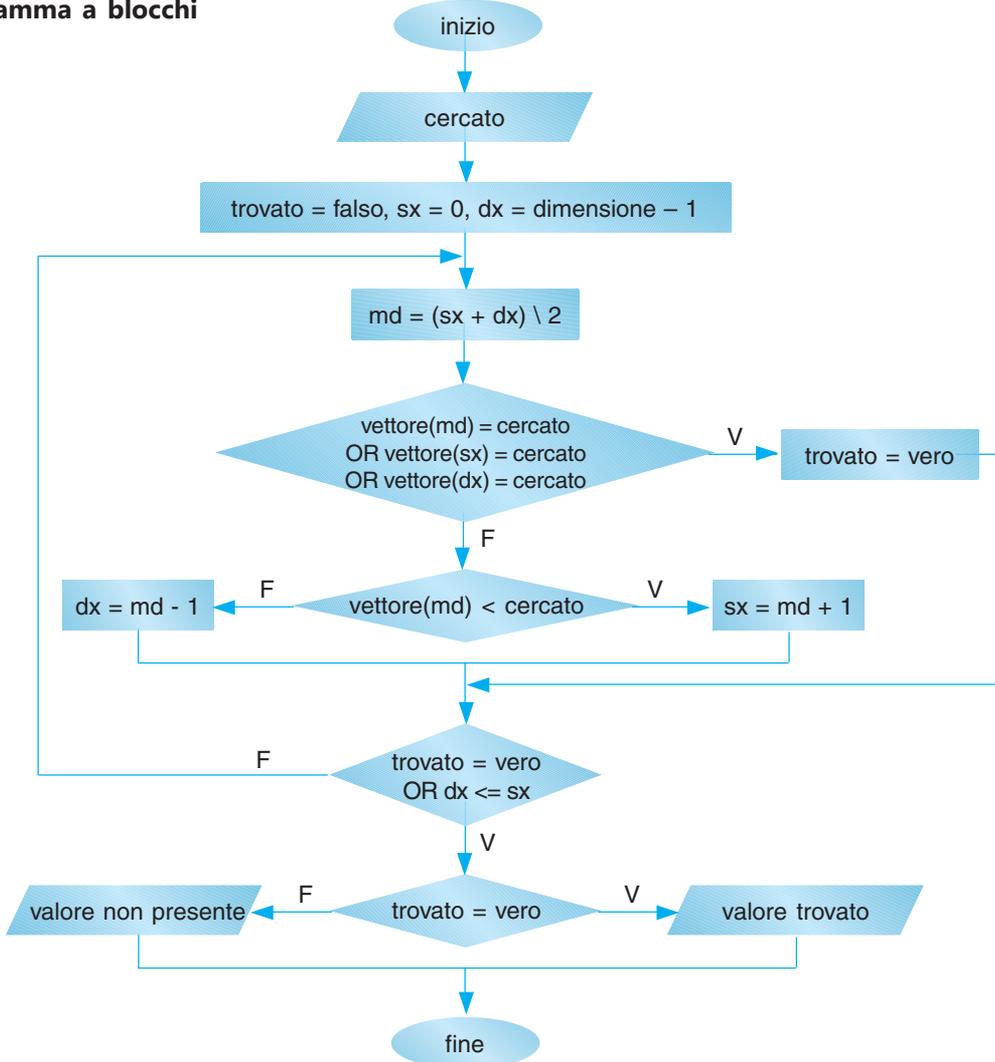
    altrimenti

        scrivi "il valore non è presente"

    fine se

fine

## Diagramma a blocchi



## Codice Visual Basic

```
Const Max As Integer = 10
Dim dimensione As Integer = 0
Dim vettore() As String
Dim trovato As Boolean

Private Sub btnCarica_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCarica.Click
    Do
        dimensione = InputBox("Inserisci la dimensione", "Chiedi dimensione")
    Loop Until dimensione >= 1 And dimensione <= Max
    ReDim vettore(dimensione - 1)
    For i = 0 To dimensione - 1
        vettore(i) = InputBox("Inserisci il " & i + 1 & "° elemento",
        "Inserisci")
    Next
End Sub
```

```

Private Sub btnTrova_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTrova.Click
    Dim cercato As String
    Dim sx, dx, md As Integer
    If txtTrova.Text = "" Then
        MessageBox.Show("Inserire il testo da ricercare", "Attenzione")
    Else
        trovato = False
        cercato = txtTrova.Text
        sx = 0
        dx = dimensione - 1
        Do
            md = (sx + dx) \ 2
            If vettore(md) = cercato Or vettore(sx) = cercato Or vettore(dx)
= cercato Then
                trovato = True
            ElseIf vettore(md) < cercato Then
                sx = md + 1
            Else
                dx = md - 1
            End If
        Loop Until trovato = True Or dx <= sx
        If trovato = True Then
            lblResult.Text = "Il valore è stato trovato"
        Else
            lblResult.Text = "Il valore non è presente"
        End If
    End If
End Sub

Private Sub btnEsci_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEsci.Click
    End
End Sub
End Class

```

Come si è detto all'inizio, l'algoritmo della ricerca dicotomica può essere applicato solo ad array ordinati. Se l'array da utilizzare non è ordinato, si deve aggiungere la procedura di ordinamento.