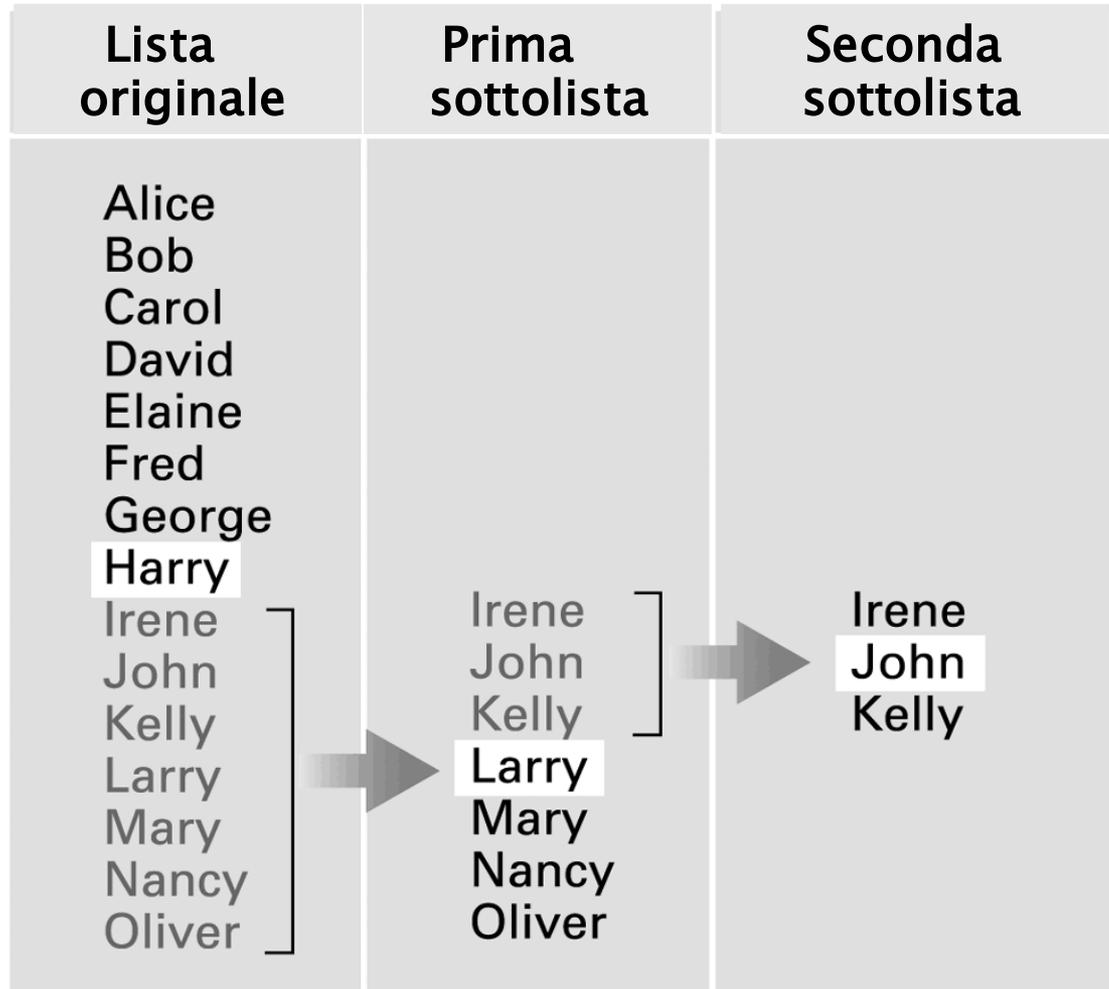


Informatica Generale

Andrea Corradini

15 - Algoritmi: Ricerca Binaria ed esercizi

La ricerca binaria: ricerca di John in una lista ordinata di nomi



Ricerca binaria in pseudocodice: un primo passo

- Cerchiamo il valore X nella sequenza Lista

```
if (Lista è vuota)
  then return false; // La ricerca è fallita
else [ElementoTest ← elemento centrale della Lista
      if (ElementoTest = X) then return true; //trovato
      else if (ElementoTest < X)
        then // X è dopo ElementoTest
              (cerca X nella parte della lista
               dopo ElementoTest; riporta il risultato)
      else // X è prima ElementoTest
            (cerca X nella parte della lista
             prima di ElementoTest; riporta il risultato)
```

]

Algoritmo di ricerca binaria in pseudocodice

- Cerchiamo il valore X nella sequenza $Lista$

```
procedure Cerca(Lista, i, j, X)
if (j < i) // Lista[i,j] è vuota
then return false; // La ricerca è fallita
else [ElementoTest ← s[(i+j)/2]
      if (ElementoTest = X) then return true; // trovato
      else if (ElementoTest < X)
          then return Cerca(Lista, (i+j)/2+1, j, X)
          else return Cerca(Lista, i, (i+j)/2-1, X)
```

- L'invocazione $Cerca(Lista, i, j, X)$ cerca X nella porzione della lista tra la posizione i e la posizione j , comprese. Quindi se la lista ha N elementi,

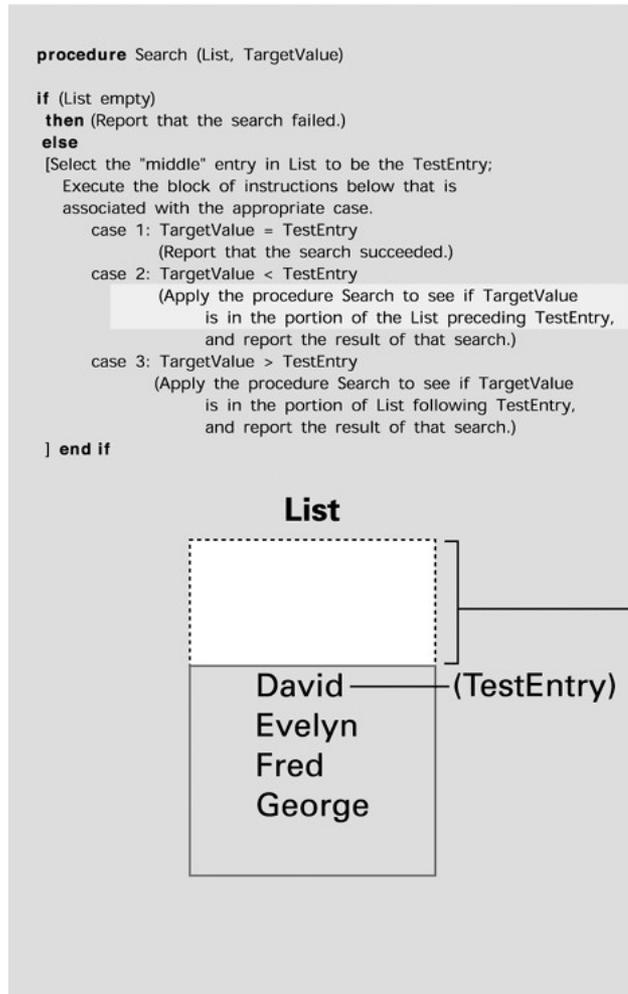
$risultato \leftarrow Cerca(Lista, 1, N, X)$

assegna $true$ a $risultato$ se e solo se X compare nella lista

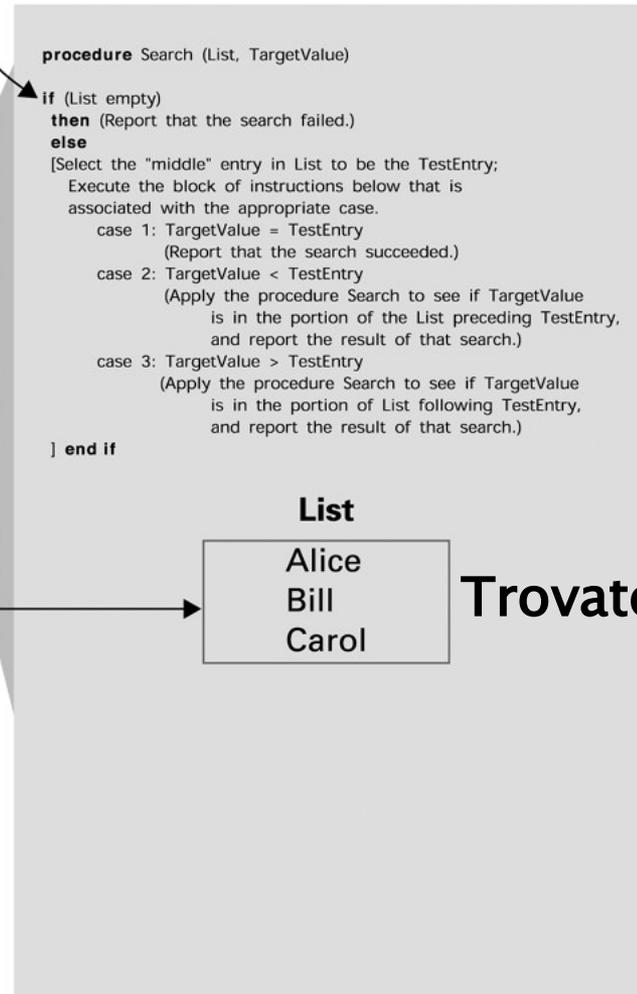
Come funziona la ricerca binaria?

Cerchiamo **Bill**
nella lista

Alice
Bill
Carol
David
Evelyn
Fred
George



We are here.



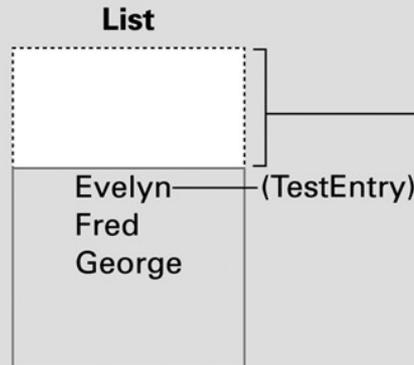
Trovato!

Esempio di ricerca binaria con fallimento

Cerchiamo **David**
nella lista

Alice
Carol
Evelyn
Fred
George

```
procedure Search (List, TargetValue)
if (List empty)
then (Report that the search failed.)
else
[Select the "middle" entry in List to be the TestEntry;
Execute the block of instructions below that is
associated with the appropriate case.
  case 1: TargetValue = TestEntry
    (Report that the search succeeded.)
  case 2: TargetValue < TestEntry
    (Apply the procedure Search to see if TargetValue
     is in the portion of the List preceding TestEntry,
     and report the result of that search.)
  case 3: TargetValue > TestEntry
    (Apply the procedure Search to see if TargetValue
     is in the portion of List following TestEntry,
     and report the result of that search.)
] end if
```



We are here.

```
procedure Search (List, TargetValue)
if (List empty)
then (Report that the search failed.)
else
[Select the "middle" entry in List to be the TestEntry;
Execute the block of instructions below that is
associated with the appropriate case.
  case 1: TargetValue = TestEntry
    (Report that the search succeeded.)
  case 2: TargetValue < TestEntry
    (Apply the procedure Search to see if TargetValue
     is in the portion of the List preceding TestEntry,
     and report the result of that search.)
  case 3: TargetValue > TestEntry
    (Apply the procedure Search to see if TargetValue
     is in the portion of List following TestEntry,
     and report the result of that search.)
] end if
```

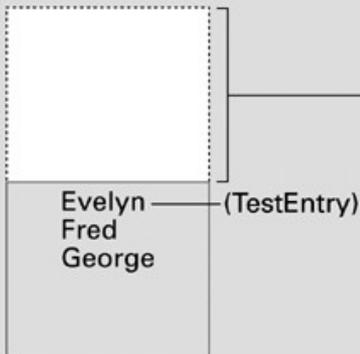


Esempio di ricerca binaria con fallimento

procedure Search (List, TargetValue)

```
if (List empty)
  then (Report that the search failed.)
else
  [Select the "middle" entry in List to be the TestEntry;
  Execute the block of instructions below that is
  associated with the appropriate case.
  case 1: TargetValue = TestEntry
    (Report that the search succeeded.)
  case 2: TargetValue < TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of the List preceding TestEntry,
    and report the result of that search.)
  case 3: TargetValue > TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of List following TestEntry,
    and report the result of that search.)
] end if
```

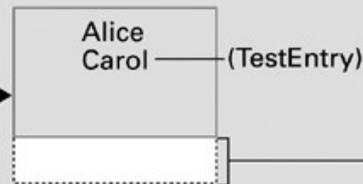
List



procedure Search (List, TargetValue)

```
if (List empty)
  then (Report that the search failed.)
else
  [Select the "middle" entry in List to be the TestEntry;
  Execute the block of instructions below that is
  associated with the appropriate case.
  case 1: TargetValue = TestEntry
    (Report that the search succeeded.)
  case 2: TargetValue < TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of the List preceding TestEntry,
    and report the result of that search.)
  case 3: TargetValue > TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of List following TestEntry,
    and report the result of that search.)
] end if
```

List



We are here.

procedure Search (List, TargetValue)

```
if (List empty)
  then (Report that the search failed.)
else
  [Select the "middle" entry in List to be the TestEntry;
  Execute the block of instructions below that is
  associated with the appropriate case.
  case 1: TargetValue = TestEntry
    (Report that the search succeeded.)
  case 2: TargetValue < TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of the List preceding TestEntry,
    and report the result of that search.)
  case 3: TargetValue > TestEntry
    (Apply the procedure Search to see if TargetValue
    is in the portion of List following TestEntry,
    and report the result of that search.)
] end if
```

List

Fallimento!

Esercizi

- Scrivere un algoritmo in pseudocodice per calcolare il massimo di una sequenza di valori. Più precisamente, scrivere una procedura **max(Lista)** che restituisce il massimo valore in **List**a, dove si assume che **List**a non sia vuota e che i suoi elementi siano confrontabili tra loro.
- *Risultato dello svolgimento in classe.* Assumiamo che la lista abbia **N** elementi, **s[1]**, **s[2]**, ..., **s[N]**. Nella variabile **massimo** ricordiamo il massimo tra gli elementi già visti. la variabile **k** contiene la posizione del prossimo elemento.

```
procedure max(Lista)
k ← 1; // posizione del prossimo elemento della lista
massimo ← s[1]; // ok perché c'è almeno un elemento
while (k ≤ N) do
    if (massimo < s[k]) // aggiorno massimo
        then massimo ← s[k];
    k = k+1;
return massimo;
```

Esercizi (2)

- Qual è il risultato di `emmecidi(18,24)`, dove `emmecidi` è la seguente procedura?

```
procedure emmecidi (X,Y)
if (X = Y)
  then return X
  else if (X > Y)
    then return emmecidi (X-Y, Y)
    else return emmecidi (X, Y-X)
```

- Scrivere una procedura ricorsiva `mult(x, y)` che restituisce il prodotto di due numeri maggiori o uguali di zero, usando la seguente definizione ricorsiva di prodotto:

$$x * 0 = 0$$

$$x * (y + 1) = x * y + x$$