

Algoritmi e Strutture Dati

Operazioni su Lista

Dr. Emanuela Merelli

Argomenti della lezione

- Tipi di Dato Astratto
 - *Lista Lineare*
 - Pila
 - Coda
- *Concetto di*
 - *Struttura dati dinamiche*

Esercizio 1

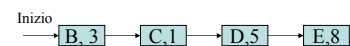
Dato un elenco di parole (testo), si vuole costruire una **Lista Ordinata** contenente un elemento per ogni distinta parola.

Il generico elemento della Lista conterrà un campo informazione (la parola) e un contatore che indicherà il numero di occorrenze della parola nell'elenco.

Esempio

Elenco

E
B
D
E
A
E
E
B
D
D
D
E
D
B
E
E
E
D



L'Algoritmo proposto è la generalizzazione dell' **Inserimento di un elemento in una lista ordinata**

Ordinamento di una lista

L'ordinamento della lista viene ottenuto inserendo nuovi elementi al posto appropriato, piuttosto che in testa alla lista

Ciò deriva dalla flessibilità con cui gli elementi possono essere inseriti in qualunque posto. Questa caratteristica non è presente nella struttura array.

Notiamo che l'equivalente ricerca binaria per array non è disponibile per le liste in generale

Inserimento in una lista Ordinata

ognuno contenente come informazione una parola e un intero n

Algoritmo

Caso 1: Lista vuota

Passo 1. L'elemento è inserito in testa

Caso 2: Lista non vuota

Passo 1. Si cerca l'elemento

Caso 2.1: L'elemento esiste nella lista

Passo 1. Si incrementa il contatore

Caso 2.2: L'elemento non esiste

Caso 2.2.1. Se il nuovo è il più piccolo

Passo 1. L'elemento è inserito in testa

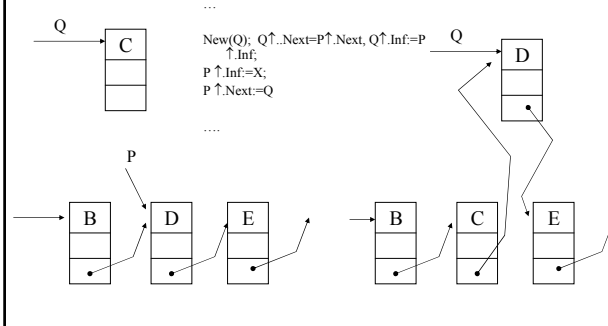
Caso 2.2.2. Se è compreso tra due valori

Passo 1. L'elemento è inserito prima dell'elemento con chiave più grande

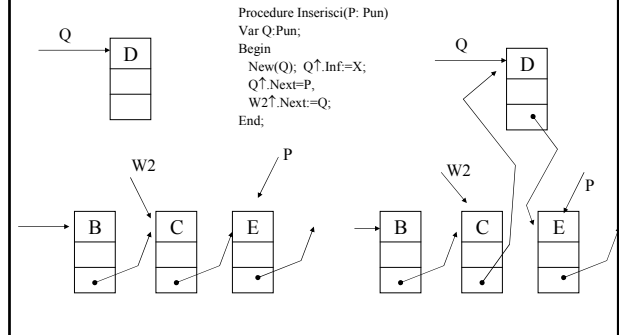
Caso 2.2.3 Se è il più grande

Passo 1. L'elemento è inserito in coda

Inserimento prima di P 1^a soluzione



Inserimento prima di P 2^a soluzione

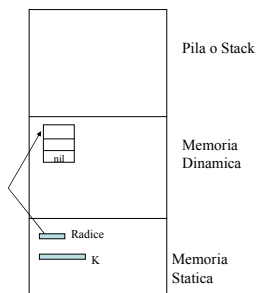


Programma Principale

```

Program Esercizio1;
Type
  stringa = ...
  Pun = ↑ Parola;
  Parola = RECORD
    Inf : Stringa;
    Conta: Integer;
    Next : Pun;
  END;
Var
  K: Stringa; Radice:Pun;
...
Begin %programma principale%
  New(Radice); Radice↑.Next:=NIL; Read(K);
  While K <> "Fine" DO Begin
    Ricerca(Radice, K)
    Read(K)
  End;
End.

```

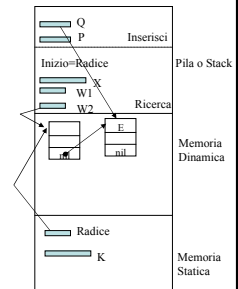


Caso 1: Lista Vuota

```

...
Var
  K: Stringa; Radice:Pun;
Procedure Ricerca ( var Inizio:Pun, X:Stringa)
  Var P1,W2 : Pun;
Begin
  W2:= Inizio; W1:=W2↑.Next;
  If W1 = NIL Then Inserisci (NIL) Else
    Begin
      While (W1↑.Inf < X And (W1↑.Next <> NIL) Do
        Begin
          W2:=W1;
          W1:=W1↑.Next;
        End;
      If W1↑.Inf = X Then W1↑.Conta:= W1↑.Conta+1
      Else Begin If W1↑.Next = NIL Then
        Begin W2:=W1; W1:=NIL; End
        Inserisci(W1); End;
    End
  End
...
Begin %programma principale%
  New(Radice); Radice↑.Next:=NIL; Read(K);
  While K <> "Fine" DO Begin
    Ricerca(Radice, K)
    Read(K); End;
  End.

```

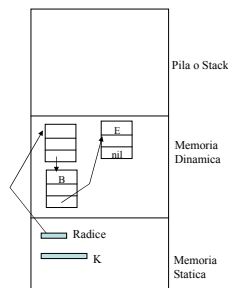


Caso 2: Lista Non Vuota Inserimento in testa

```

...
Var
  K: Stringa; Radice:Pun;
Procedure Ricerca ( var Inizio:Pun, X:Stringa)
  Var P1,W2 : Pun;
Begin
  W2:= Inizio; W1:=W2↑.Next;
  If W1 = NIL Then Inserisci (NIL) Else
    Begin
      While (W1↑.Inf < X And (W1↑.Next <> NIL) Do
        Begin
          W2:=W1;
          W1:=W1↑.Next;
        End;
      If W1↑.Inf = X Then W1↑.Conta:= W1↑.Conta+1
      Else Begin If W1↑.Next = NIL Then
        Begin W2:=W1; W1:=NIL; End
        Inserisci(W1); End;
    End
  End
...
Begin %programma principale%
  New(Radice); Radice↑.Next:=NIL; Read(K);
  While K <> "Fine" DO Begin
    Ricerca(Radice, K)
    Read(K); End;
  End.

```

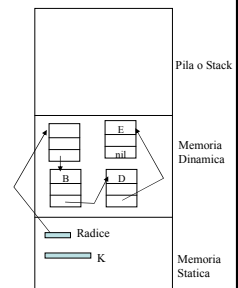


Caso 3: Lista Non Vuota Inserimento in testa

```

...
Var
  K: Stringa; Radice:Pun;
Procedure Ricerca ( var Inizio:Pun, X:Stringa)
  Var P1,W2 : Pun;
Begin
  W2:= Inizio; W1:=W2↑.Next;
  If W1 = NIL Then Inserisci (NIL) Else
    Begin
      While (W1↑.Inf < X And (W1↑.Next <> NIL) Do
        Begin
          W2:=W1;
          W1:=W1↑.Next;
        End;
      If W1↑.Inf = X Then W1↑.Conta:= W1↑.Conta+1
      Else Begin If W1↑.Next = NIL Then
        Begin W2:=W1; W1:=NIL; End
        Inserisci(W1); End;
    End
  End
...
Begin %programma principale%
  New(Radice); Radice↑.Next:=NIL; Read(K);
  While K <> "Fine" DO Begin
    Ricerca(Radice, K)
    Read(K); End;
  End.

```



Ordinamento di una Lista

Esercizio

Data una lista di N elementi ordinare gli elementi, in ordine decrescente del campo informazione supposto essere di tipo Integer

Complessità?