

Introduzione agli Algoritmi

Esercizi su Liste

Dr. Emanuela Merelli

Argomenti della lezione

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

Lista Lineare

è un insieme ordinato di n elementi, tutti dello stesso tipo, ogni elemento ha un predecessore e un successore

Proprietà:

1. X_1 è il primo elemento della lista
2. X_i è preceduto da x_{i-1} e seguito da x_{i+1}
3. X_n è l'ultimo elemento della lista
4. N = lunghezza della lista
5. $N=0 \Rightarrow$ Lista Vuota



$L(A, B, C, D)$

Relazione di precedenza: per accedere ad un elemento si deve accedere a tutti quelli che lo precedono

Lista Lineare come Tipo di Dato Astratto

Dominio dei valori

- Insieme dinamico basato su un elemento omogeneo

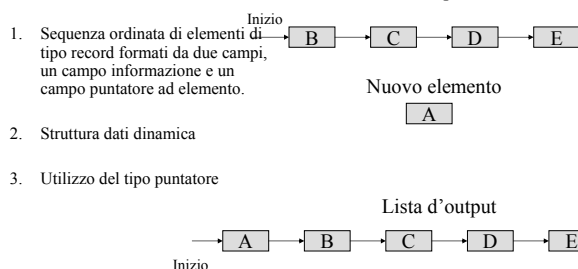
Operazioni su Lista Lineare

- Inserire un elemento in testa
- Inserire un elemento in coda
- Inserire un elemento dopo un elemento puntato da P
- Creare una lista con N elementi
-

Operazioni su Liste Lineari

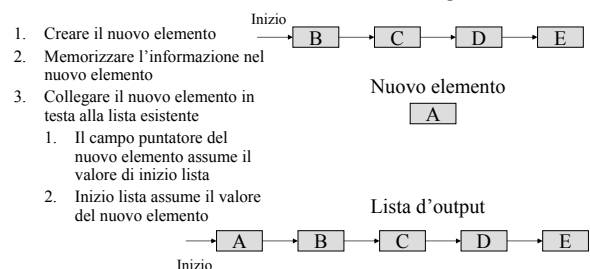
Inserimento di un elemento in Testa alla Lista

Struttura Dati



Inserimento di un elemento in testa

Algoritmo



Inserimento di un elemento in testa

```

Type
  T = RECORD
    Inf : Integer;
    Next : Tp;
  END;
Pun = ↑ T;
Var
  Primo: Pun; A: T;
Procedura Testa (Var Inizio: Pun, X: Integer)
  Var Q: Pun;
Begin
  New(Q);
  Q↑.Inf := X;
  Q↑.Next := Inizio;
  Inizio := Q;
END;

```

Lista d'input
Inizio → B → C → D → E

Nuovo elemento
A

Lista d'output
Inizio → A → B → C → D → E

Homework

Definire una Funzione Testa di tipo puntatore che prenda in input un valore X e restituisca Testa con X come primo elemento

Eliminazione dell'elemento in testa ad una lista

```

Type
  T = RECORD
    Inf : Integer;
    Next : Tp;
  END;
Pun = ↑ T;
Var
  Primo: Pun; A: T;
Procedura Testa (Var Inizio: Pun, X: Integer)
  Var Q: Pun;
Begin
  New(Q);
  Q↑.Inf := X;
  Q↑.Next := Inizio;
  Inizio := Q;
END;

```

Lista d'input
Inizio → A → B → C → D → E

Elemento eliminato
A

Lista d'output
Inizio → B → C → D → E

Scansione di una lista lineare

```

...
Procedura Scansione (Var Inizio: Pun)
Begin
  WHILE Inizio <> NIL DO
    Begin
      Writeln(Inizio.Inf);
      Inizio := Inizio↑.Next;
    End;
  End;

```



Inserimento di un elemento in coda

```

...
Procedura Coda (Var Inizio: Pun, X: Integer)
  Var P, Q: Pun;
Begin
  New(Q);
  Q↑.Next := NIL;
  Q↑.Inf := X;
  P := Inizio;
  IF Inizio = NIL THEN Inizio := Q
  ELSE Begin
    WHILE P↑.Next <> NIL DO
      P := P↑.Next;
    P↑.Next := Q;
  End;
END;

```

Lista d'input
Inizio → A → B → C → D

Nuovo elemento
E

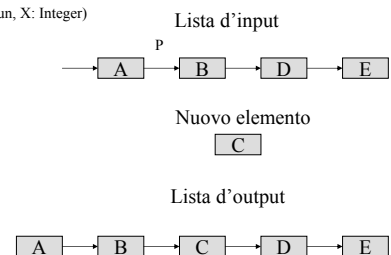
Lista d'output
A → B → C → D → E

Inserimento di un elemento dopo un elemento puntato da P

```

...
Procedura Inserisci (Var P: Pun, X: Integer)
  Var Q: Pun;
Begin
  New(Q);
  Q↑.Inf := X;
  Q↑.Pun := P↑.Next;
  Q↑.Next := P;
END;

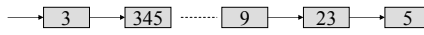
```



Creazione di una lista di N elementi

...

```
Procedura Crea (Var Inizio: Pun, N: Integer)
Var P, Q: Pun;
Begin
  Inizio:= NIL;
  WHILE N>0 DO
    Begin
      New(Q);
      Q↑.Next:= Inizio;
      Inizio:= Q;
      Q↑.Inf:=N;
      N:= N-1
    End;
  END;
```



Esercizio 1

Data un elenco di parole, 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.

Inserimento in Liste Ordinate

```
Type
stringa = ...
Pun = ↑ Parola;
Parola = RECORD
  Inf: Stringa;
  Conta: Integer;
  Next: Pun;
END;

Var
  K: Stringa; Radice:Pun;
Procedura Ricerca (Var Inizio: Pun, X: stringa)
Var W1,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
```

```
Procedura Inserisci (W:Pun)
```

```
Var W3: Pun;
Begin
  New (W3);
  W3↑.Inf:= X; W3 ↑.Conta:= 1;
  W3↑.Next:=W
  W2↑.Next:=W3
End;
```

```
Begin %programma principale%
  New(Radice); Radice ↑.Next:=NIL; Read(K);
  While K <> "Fine" DO Begin
    Ricerca(Radice, K)
    Read(K)
  End;
End.
```