

Laboratorio di Informatica

Lezione 6: Le strutture a lista



30-01-2004

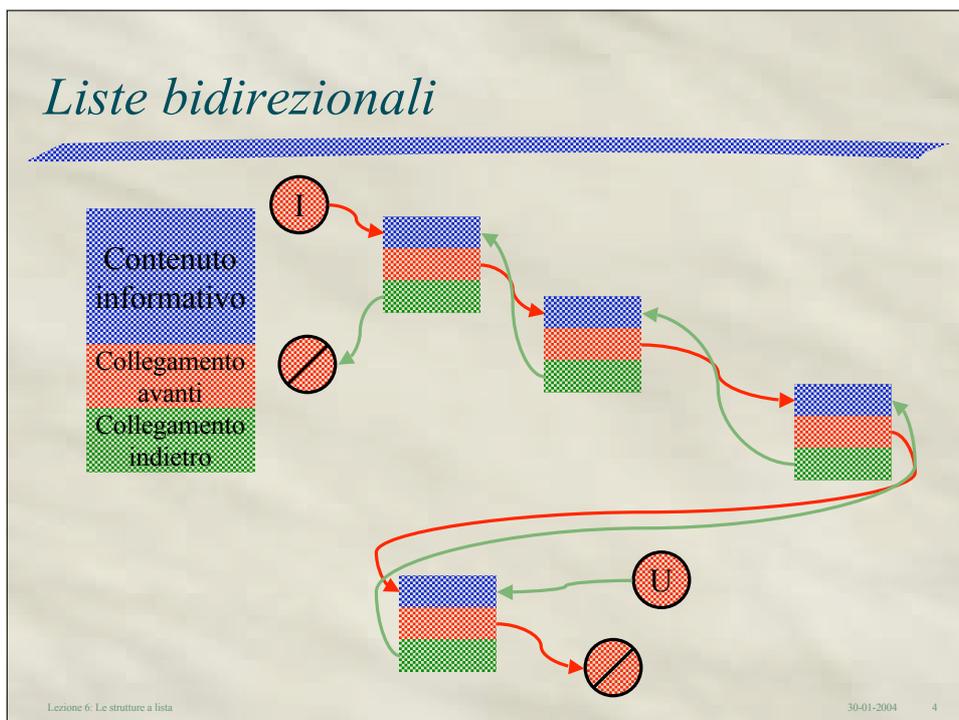
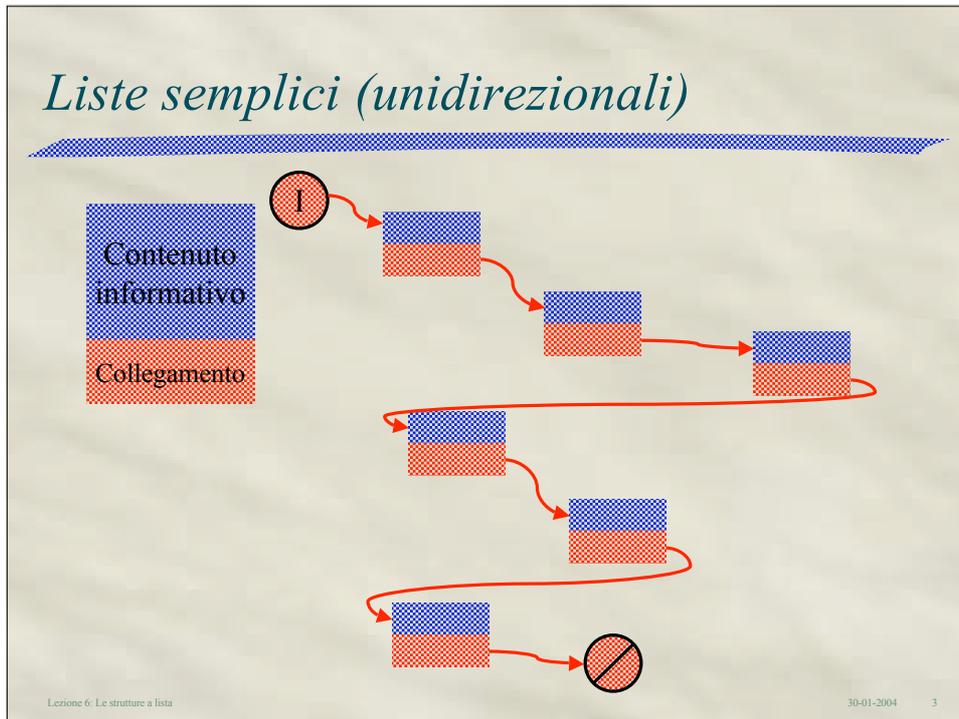
Strutture a lista

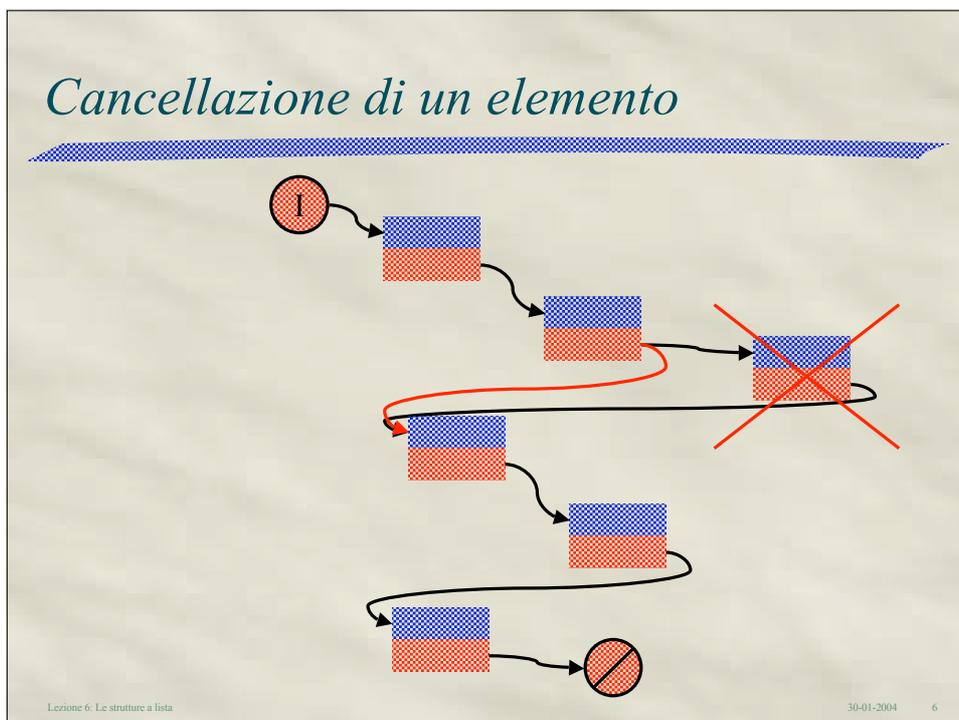
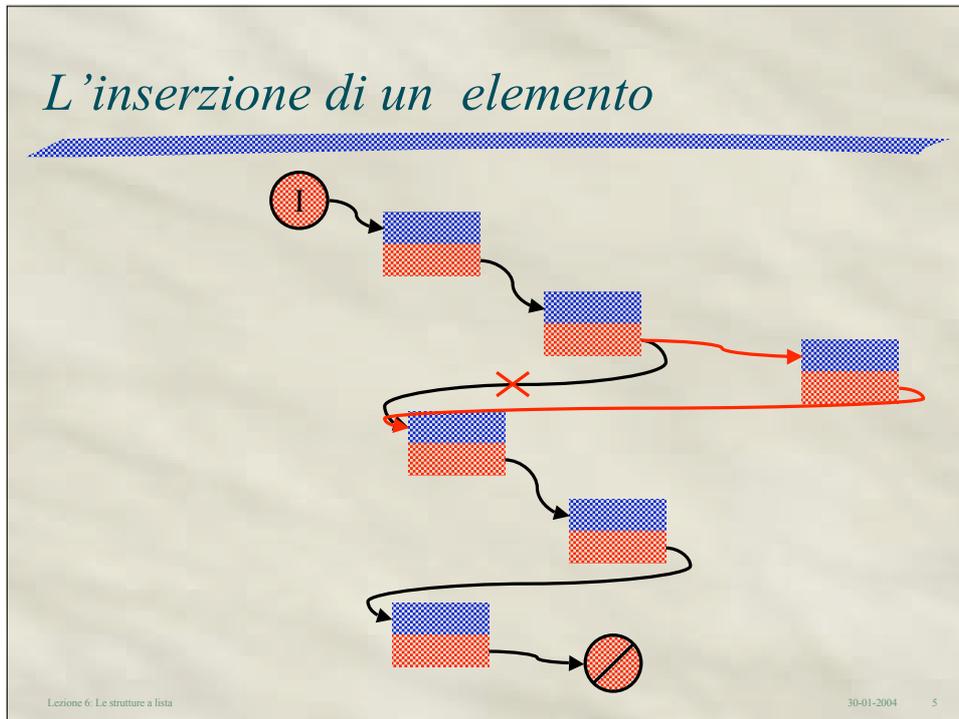


- ⇒ I vettori contengono informazioni memorizzate in maniera sequenziale
 - Devono occupare celle di memoria contigue
 - La loro dimensione deve essere decisa a priori
 - Ordinare un vettore implica spostarne i contenuti (almeno una parte, se si usa un vettore a indice)
- ⇒ Le liste eliminano tutti questi problemi, con tre svantaggi:
 - Maggiore occupazione di memoria
 - Impossibilità di accesso diretto a un determinato elemento
 - Possibilità di perdita di dati in caso di errori

Lezione 6: Le strutture a lista

30-01-2004 2





Prerequisiti (1)

```
typedef struct node
{
    int key;
    // int info;
    struct node *psnNext;
}NODE, *PNODE;
```

Lezione 6: Le strutture a lista

30-01-2004 7

Prerequisiti (2)

```
int main()
{
    int iX;
    PNODE psnStart, psnTemp;
    psnStart=NULL;
    printf("Introdurre una sequenza di interi, "
           "seguita da #:\n");
```

Lezione 6: Le strutture a lista

30-01-2004 8

Prerequisiti (2)

```
while (scanf("%d", &iX))
{
    psnTemp=psnStart;
    psnStart=(PNODE) malloc (sizeof(NODE));
    if (psnStart == NULL)
    {
        printf ("Memoria insufficiente");
        exit (1); //MOLTO IMPORTANTE!
    }
    psnStart->key=iX;
    psnStart->psnNext=psnTemp;
}
```

Lezione 6: Le strutture a lista

30-01-2004 9

Prerequisiti (3)

```
printf("\nIn ordine inverso, sono stati "\n"
      "letti i seguenti numeri:\n");
for (psnTemp=psnStart; psnTemp!=NULL;
     psnTemp=psnTemp->psnNext)
    printf ("%5d\n", psnTemp->key);
}
```

Lezione 6: Le strutture a lista

30-01-2004 10

Liste con sentinella

- ⇒ Il problema delle liste è che servirebbero algoritmi diversi se la lista è vuota o se contiene elementi
- ⇒ Utilizziamo liste “con sentinella”, ovvero liste che contengono sempre almeno un elemento

Lezione 6: Le strutture a lista

30-01-2004 11

Creazione della lista con sentinella

```
PNODE creanodo ()
{
    PNODE psn;
    psn=(PNODE) malloc(sizeof(NODE));
    if (psn == NULL)
    {
        printf("Memoria insufficiente \n"); exit (1);
    }
    return psn;
}

PNODE psnTemp, psnStart, psnEnd;

psnStart = psnEnd = creanodo();
```

Lezione 6: Le strutture a lista

30-01-2004 12

Inserimento in cima alla lista

```
psnTemp = psnStart;  
psnStart = creanodo();  
psnStart->psnNext = psnTemp;  
psnStart->key = iX;
```

Inserimento in fondo alla lista

```
psnTemp = psnEnd;  
psnEnd= creanodo();  
psnTemp->psnNext = psnEnd;  
psnTemp->key = iX;
```