

Laboratorio di Informatica

Lezione 9: Operazioni secondarie

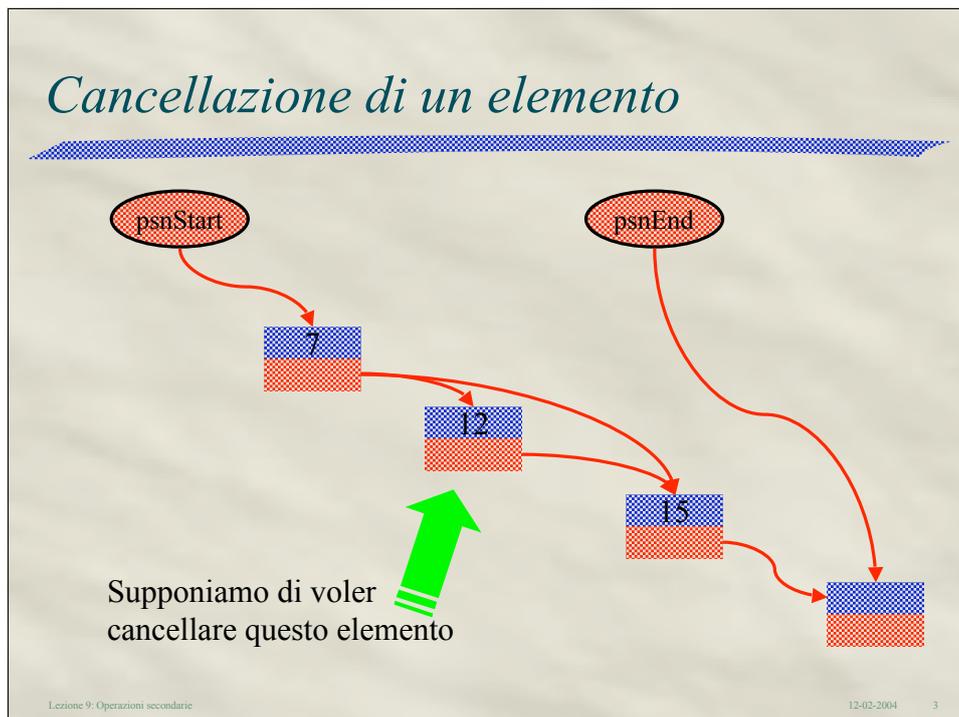
12-02-2004

Cancellazione di un elemento di una lista

- ⇒ Anche la cancellazione sarebbe un'operazione semplice...
- ⇒ Basterebbe “staccare” un elemento e ricollegare il puntatore...

Lezione 9: Operazioni secondarie

12-02-2004 2



Ma come al solito...

- ⇒ Non disponiamo del puntatore all'elemento precedente della lista. Quindi non possiamo aggiornare il puntatore.
- ⇒ Usiamo il trucco già visto per l'inserzione:
 - Copiamo il contenuto dell'elemento successivo nell'elemento puntato da psn
 - Cancelliamo l'elemento successivo a quello puntato da psn

Lezione 9: Operazioni secondarie 12-02-2004 4

In pratica:

```

/* cancellazione del nodo puntato da psn */

PNODE psnQ; //un puntatore temporaneo

psnQ = psn->psnNext; //psnQ punta al nodo successivo

if (psnQ == psnEnd)
    psnEnd = psn; //se psn puntava al penultimo nodo,
                //basta aggiustare la sentinella
else
    *psn = *psnQ; //copiamo l'intero nodo successivo
                //puntatore compreso

free (psnQ); //e non dimentichiamo di liberarlo!

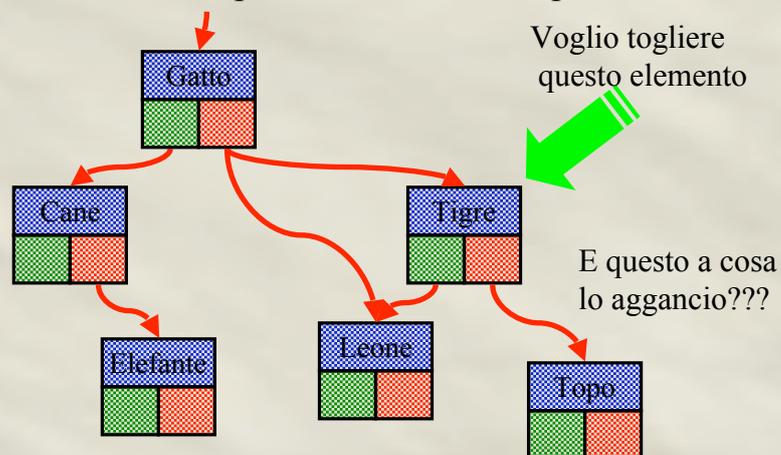
```

Lezione 9: Operazioni secondarie

12-02-2004 5

Cancellazione di un nodo di un albero:

⇒ Questa invece è un'operazione molto complessa:



Lezione 9: Operazioni secondarie

12-02-2004 6

Giusto per la cronaca...

```

void delnode(PTRPTRNODO ppsn)
{
    PTRPTRNODO ppsnQQ;
    PTRNODO psn, psnQ;
    if (*ppsn != NULL)
    {
        psn = *ppsn;
        if (psn->psnRight == NULL)
        {
            *ppsn = psn->psnLeft;
            free(psn);
        }
        else
        {
            if (psn->psnLeft == NULL)
            {
                *ppsn = psn->psnRight;
                free(psn);
            }
            else
            {
                ppsnQQ = &psn->psnLeft;
                while ((*ppsnQQ)->psnRight != NULL)
                    ppsnQQ = &(*ppsnQQ)->psnRight;
                psnQ = *ppsnQQ; /* elemento più dx del sottoalbero di sx */
                                /* cioè il più grande tra gli inferiori */
                *ppsnQQ = psnQ->psnLeft; /* ricollega il sottoalbero sx di psnQ se esiste */
                psn->iNum = psnQ->iNum; /* sostituisco solo il dato non il nodo */
                free(psnQ);
            }
        }
    }
}

```

Lezione 9: Operazioni secondarie

12-02-2004 7

Invece è semplice cancellare un intero sottoalbero:

```

void deltree (PTRPTRNODO ppsn)
{
    if (*ppsn != NULL)
    {
        deltree (&(*ppsn)->psnLeft);
        deltree (&(*ppsn)->psnRight);
        free(*ppsn);
        *ppsn = NULL;
    }
}

```

Ma attenzione ai puntatori doppi!

Lezione 9: Operazioni secondarie

12-02-2004 8

Ordinamento di una lista

- ⇒ Se dobbiamo mettere in ordine una lista disordinata, l'algoritmo giusto è
 - Estrarre il primo elemento dalla lista ordinata
 - Inserirlo in una seconda lista, al posto giusto
 - Ripetere finché la prima lista è vuota
- ⇒ **Attenzione:** non occorre creare nuovi elementi: basta sistemare i puntatori degli elementi che ci sono già.

Altri esercizi:

- ⇒ Analizzare l'algoritmo per la cancellazione di un nodo
- ⇒ Verificare che sia corretto
- ⇒ Utilizzarlo per cancellare una parola dall'albero costruito la volta scorsa.