

Pseudocodice

E. Liste lineari

E.1. Ricerca in base alla posizione

```
Funzione elemento (Lista  $\ell$ , intero  $i$ )  $\rightarrow$  Nodo  
/* Restituisce il riferimento all' $i$ -esimo nodo della lista o null se la lista  
   è piú corta ( $i$  nodi sono contati a partire da 0). */  
 $p \leftarrow \ell$   
while  $p \neq \text{null}$  and  $i > 0$  do  
  |  $p \leftarrow p.\text{pros}$   
  |  $i \leftarrow i - 1$   
return  $p$ 
```

E.2. Ricerca in base alla chiave

```
Funzione trova (Lista  $\ell$ , tipoChiave  $k$ )  $\rightarrow$  Nodo  
/* Restituisce il riferimento al primo nodo della lista la cui chiave  
   coincide con il valore fornito tramite il parametro  $k$ .  
   Se tale chiave non è presente restituisce null. */  
 $p \leftarrow \ell$   
while  $p \neq \text{null}$  and  $p.\text{chiave} \neq k$  do  
  |  $p \leftarrow p.\text{pros}$   
return  $p$ 
```

E.3. Ricerca in una lista ordinata in base alla chiave

```
Funzione trova (ListaOrdinata  $\ell$ , tipoChiave  $k$ )  $\rightarrow$  Nodo  
/* Restituisce il riferimento al primo nodo della lista la cui chiave  
   coincide con il valore fornito tramite il parametro  $k$ .  
   Se tale chiave non è presente restituisce null. */  
 $p \leftarrow \ell$   
while  $p \neq \text{null}$  and  $p.\text{chiave} < k$  do  
  |  $p \leftarrow p.\text{pros}$   
if  $p = \text{null}$  or  $p.\text{chiave} > k$  then  
  | return null  
else  
  | return  $p$ 
```

E.4. Inserimento in una lista ordinata

Funzione *inserisci* (*ListaOrdinata* ℓ , *elemento* d) \rightarrow *ListaOrdinata*

/* Inserisce nella lista ordinata un nuovo nodo e restituisce il riferimento alla lista così modificata. */

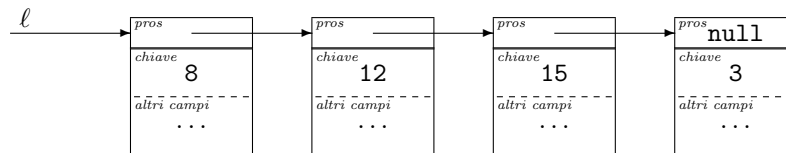
 $k \leftarrow d.chiave$ $p \leftarrow \ell$ // riferimento all'elemento in esame (a partire dal primo) $prec \leftarrow null$ // riferimento all'elemento precedente quello in esame**while** $p \neq null$ **and** $p.chiave < k$ **do**┌ $prec \leftarrow p$ └ $p \leftarrow p.pros$ $r \leftarrow$ riferimento a nuovo nodo $r.chiave \leftarrow k$ $r.altri\ campi \leftarrow d.altri\ campi$ $r.pros \leftarrow p$ **if** $prec = null$ **then**┌ $\ell \leftarrow r$ **else**┌ $prec.pros \leftarrow r$ **return** ℓ

E.5. Cancellazione da una lista ordinata

Funzione *cancella* (*ListaOrdinata* ℓ , *tipoChiave* k) \rightarrow *ListaOrdinata*/* Cancella dalla lista ordinata il primo nodo di chiave k e restituisce il riferimento alla lista così modificata. */ $p \leftarrow \ell$ $prec \leftarrow null$ **while** $p \neq null$ **and** $p.chiave < k$ **do**┌ $prec \leftarrow p$ └ $p \leftarrow p.pros$ **if** $p \neq null$ **and** $p.chiave = k$ **then**┌ **if** $prec = null$ **then**└┌ $\ell \leftarrow p.pros$ └└ **else**└└┌ $prec.pros \leftarrow p.pros$ **return** ℓ

Note

- Una lista è formata da una collezione di nodi collegati linearmente tra loro tramite puntatori. Il campo *pros* di ciascun nodo contiene il riferimento al nodo successivo. Gli altri campi contengono le informazioni che devono essere memorizzate nel nodo. Tra questi vi è un campo, che chiamiamo *chiave*, rispetto al quale vengono effettuate le operazioni di ricerca. Inoltre, nel caso delle liste ordinate, il campo *chiave* viene utilizzato per determinare l'ordine tra i nodi. Indichiamo con *tipoChiave* il tipo del campo *chiave* (nella figura seguente il campo *chiave* è intero). Si accede alla lista tramite un riferimento al primo nodo (nella figura indicato con ℓ).



- Nello pseudocodice *Lista* e *Nodo* rappresentano tipi riferimento a liste e nodi di liste. A livello implementativo sono di solito puntatori a elementi dello stesso tipo (nodo di una lista). Nelle intestazioni delle funzioni vengono distinti per enfatizzare il significato differente. Ad esempio la Funzione E.1 riceve come primo parametro una lista (o meglio il riferimento alla lista) e ha lo scopo di restituire un nodo (o meglio il riferimento ad esso, che in realtà è anche il riferimento alla parte di lista che inizia da quel nodo). Lo stesso vale per *ListaOrdinata*.
- Nella Funzione E.4, *elemento* indica il tipo dei dati che vengono memorizzati nella lista (quindi, con riferimento alla figura, un dato di tipo *elemento* contiene *chiave* e *altri campi*).
- Nella Funzione E.4, se vi sono già nodi con la stessa chiave del nodo da inserire, l'inserimento avviene prima di essi. Con una semplice modifica si può fare in modo che avvenga dopo.
- Le Funzioni E.4 e E.5 possono essere sostituite da procedure, se si fornisce il primo argomento *per riferimento*. La procedura deve cioè ricevere il riferimento alla variabile che contiene il riferimento iniziale alla lista. In tal modo tale variabile risulta effettivamente modificata nell'ambiente chiamante.
- È utile studiare l'implementazione concreta in un linguaggio di programmazione (ad esempio in C utilizzando i puntatori).
- Implementando la cancellazione è si deve rilasciare l'area di memoria utilizzata dal nodo cancellato.