

in rosso di fianco ad ogni nodo mettiamo la sua posizione nell'ordine infisso,. Si osservi che le posizioni iniziano da 1. Quindi, se  $k=2$  la lista da costruire sarà: 2->0->2->1->1 che corrisponde ai nodi nelle posizioni 2, 4, 6, 8, e 10 dell'ordine infisso. Quando pickric\_infix crea l'ultimo nodo della lista, chiamiamolo a, il nodo 10 sarà il nodo corrente e l'invocazione ricorsiva deve ritornare al nodo 11 con  $n=1$  che indica che

11 è il prossimo nodo incontrato nell'ordine infisso dopo aver creato a. Quando pickric\_infix lascia il nodo 11 per tornare alla radice R, n deve essere 2 come richiede la POST:  $2-1=1$  e infatti un solo nodo nell'ordinamento infisso è stato attraversato dopo aver creato a. Inoltre  $n=2$  dice che il prossimo nodo che verrà attraversato secondo l'ordine infisso dopo la radice (dobbiamo immaginare che R sia un nodo interno di un albero più grande) sarà il secondo nodo nell'ordine infisso dopo aver creato a. Insomma n è il contatore che ci dice quando dovremo creare il prossimo nodo e appenderlo in fondo alla lista che sta costruendo.

Con lo stesso albero consideriamo che succede con  $k=3$ : la lista corrisponderà ai nodi 3, 6, 9 dell'albero e quindi sarà:  $3 \rightarrow 2 \rightarrow 2$ , restano 2 nodi nell'albero rispetto all'ordine infisso (il 10 e l'11) e quindi  $n=3$ .

Se  $k=4$ , allora la lista corrisponderà ai nodi 4 e 8 e quindi sarà:  $0 \rightarrow 1$  mentre  $n=4$  visto che restano 3 nodi (9, 10 e 11) nell'albero secondo l'ordine infisso.

**3) (8 punti)** Il terzo esercizio richiede di scrivere una funzione **iterativa** che, data la lista prodotta dalla funzione pickric\_infix dell'esercizio (2), elimina dalla lista stessa i nodi con campo info ripetuti lasciando un solo nodo per ogni valore di campo info presente. Il nodo che resta per ogni valore di campo info deve essere il primo con quel valore nella sequenza originale

**Esempio:** se la sequenza è:  $Q=2 \rightarrow 0 \rightarrow 3 \rightarrow 0 \rightarrow 2 \rightarrow 1 \rightarrow 0$ , allora si vuole ottenere la sequenza  $2 \rightarrow 0 \rightarrow 3 \rightarrow 1$  dove vengono tolti dalla sequenza x il quarto, il quinto e l'ultimo nodo. I nodi tolti formano la lista  $2 \rightarrow 0 \rightarrow 0$ . Osservare che la lista dei nodi tolti è prodotta considerando i nodi di Q dall'inizio (iniziando quindi da 2) e inserendo nella lista dei nodi tolti gli altri nodi con lo stesso info che si trovano alla sua destra. Quindi partendo dal nodo iniziale con  $\text{info}=2$ , va eliminato il 2 in quinta posizione. Poi si considera il secondo nodo, con  $\text{info}=0$  e si eliminano gli altri 2 nodi con  $\text{info}=0$ , cioè il quarto e l'ultimo. La lista dei nodi tolti sarà quindi quella detta prima,  $2 \rightarrow 0 \rightarrow 0$ , mentre della lista iniziale resterà  $2 \rightarrow 0 \rightarrow 3 \rightarrow 1$ .

**Aiuto:** visto che si vuole scorrere iterativamente una lista per eliminare alcuni suoi nodi, conviene introdurre un array dinamico di puntatori ai nodi della lista. In questo modo ci si può muovere liberamente sui nodi della lista. Si osservi però che quando si eliminano i nodi della lista, ci saranno elementi dell'array che non puntano più a nulla. Si dovrà fare molta attenzione a gestire questi casi in modo appropriato. Si osservi che la costruzione dell'array di puntatori che viene consigliata, in un certo senso è una simulazione dello stack dei record di attivazione che verrebbe costruito durante l'esecuzione di una funzione ricorsiva che scorra la lista per mezzo della ricorsione.

La funzione da fare ha il seguente prototipo: `FIFO tieni_primo(nodo*& Q)` e deve rispettare le seguenti PRE e POST:

$\text{PRE}=(L(Q) \text{ è corretta}, vL(Q)=L(Q))$

$\text{POST}=(L(Q) \text{ è } vL(Q) \text{ in cui sono stati eliminati i nodi che hanno un campo info tale che un nodo alla loro sinistra abbia lo stesso campo info. Quindi per ogni valore info resta solo il primo nodo con quel valore. La lista dei nodi eliminati va gestita da un valore FIFO restituito col return})$

**Correttezza :**

- 1) **(5 punti)** delineare la prova induttiva della correttezza di pickric\_infix
- 2) **(3 punti)** scrivere l'invariante del ciclo principale di tieni\_primo.