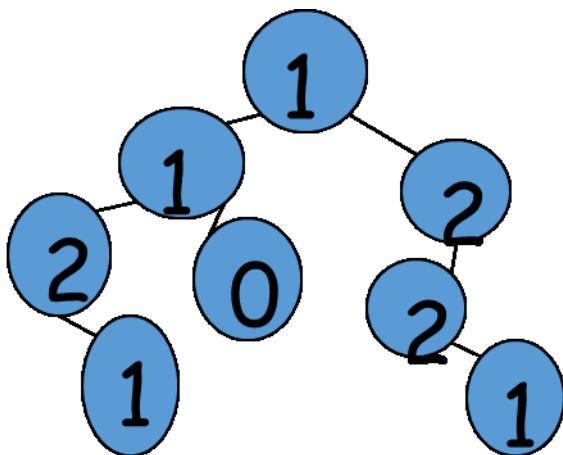


Esercizio 3 del 20/5/2021

Abbiamo visto a lezione che un percorso di un albero binario determina un ordine totale dei suoi nodi. Per esempio per il seguente albero,



il percorso prefisso determina il seguente ordine: 1 1 2 1 0 2 2 1, mentre il percorso infisso determina questo ordine: 2 1 1 0 1 2 1 2.

Sono richieste funzioni ricorsive capaci di percorrere un qualsiasi albero binario in modo prefisso e infisso, stampando il campo info di un nodo ogni k nodi attraversati, per $k > 0$.

Esempio 1. Consideriamo l'albero raffigurato sopra, prendiamo l'ordine prefisso e $k=3$. Allora la funzione desiderata dovrebbe stampare 2 2 (che sono i nodi in terza e sesta posizione nella sequenza dei nodi dell'albero che corrisponde all'ordine prefisso). Per sicurezza specifichiamo il cammino che congiunge i 2 nodi stampati alla radice: [0,0] e [1].

Se prendiamo invece l'ordine infisso e $k=4$ allora la funzione desiderata dovrebbe stampare: 0 2 (che sono ovviamente i nodi in quarta e ottava posizione nella sequenza dei nodi dell'albero che corrisponde all'ordine infisso). I cammini dei 2 nodi stampati sono: [0,1] e [1].

Si chiede di scrivere 2 funzioni ricorsive, una che faccia quanto appena spiegato seguendo l'ordine prefisso e l'altra seguendo quello infisso. Le 2 funzioni hanno segnatura e PRE e POST uguali e quindi specifichiamo queste cose solo per la funzione stampaASPre (dove AS sta per A Salti e Pre sta per prefisso) che segue l'ordine prefisso.

PRE=(albero(r) ben formato, $0 \leq n \leq k$, $k > 0$)

int stampaASPre(nodo*r, int n, int k)

POST=(considerando i nodi di albero(r) in ordine prefisso, salta n nodi e poi stampa quello successivo e dopo ne salta k-1 e poi stampa il successivo, restituisce col return un intero m tale che $(k-1)-m$ è il numero di nodi che sono stati saltati dopo l'ultimo stampato).

Per spiegare il significato dell'intero che le funzioni devono restituire col return, facciamo riferimento all'Esempio 1.

Esempio 2. Nell'Esempio 1 abbiamo visto che con $k=3$, nella sequenza dei nodi dell'albero che corrisponde al percorso prefisso, si stampano il terzo e il sesto nodo: 1 1 2 1 0 2 2 1 che abbiamo sottolineato. Dopo il sesto nodo nel vengono saltati 2 prima di terminare l'albero. Questo significa che se la sequenza di nodi continuasse, dovremmo stampare il primo nodo che verrebbe subito dopo la sequenza. Per segnalare

questo fatto, la funzione stampaASPre dovrebbe restituire col return 0. Si osservi che 0 soddisfa la POST infatti : $(3-1)-0=2$, dove 2 è il numero di nodi dell'albero saltati dopo l'ultimo stampato.

Consideriamo adesso l'ordine infisso con $k=4$. Nell'Esempio 1 abbiamo visto che in questo caso verrebbero stampati il quarto e l'ottavo della sequenza che corrisponde all'ordine infisso: 2 1 1 0 1 2 1 2, per cui la funzione stampaASInf dovrebbe restituire 3 col return, infatti $(4-1)-3=0$ che corrisponde al fatto che si è stampato l'ultimo nodo dell'albero e quindi, se la sequenza continuasse, si dovrebbero saltare 3 nodi prima della prossima stampa.

L'intero restituito col return serve, una volta terminata la stampa con salti in un sottoalbero, per continuarla correttamente nel resto dell'albero.

Correttezza: dimostrare induttivamente la correttezza di una delle 2 funzioni.