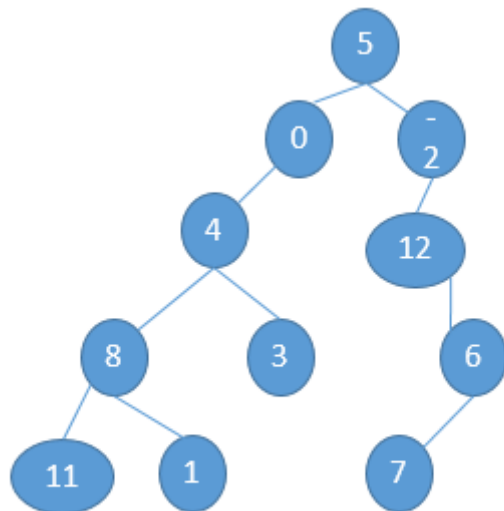


Esercizio del 23/5/2018

Si tratta di costruire un albero bilanciato con n interi letti da cin, come nell'esercizio a tempo del 17/5. Si vuole poi percorrere l'albero così costruito in ordine prefisso, stampando i campi info dei nodi attraversati, ma solo 1 nodo ogni k nodi attraversati, per un dato $k > 0$.

Esempio.

Se l'albero da attraversare è il seguente:



Allora l'ordine prefisso dei suoi nodi è 5 0 4 8 11 1 3 2 12 6 7. Per cui se $k=2$, allora vogliamo stampare: 0 8 1 2 6. Se $k=4$ vogliamo stampare: 8 2.

Queste stampe devono essere eseguite da una funzione ricorsiva `stampa_a_salti` che rispetta le seguenti specifiche:

PRE= (albero(r) è ben formato, $k > 0$, $1 \leq n \leq k$)

int `stampa_a_salti`(nodo* r , int k , int n)

POST=(stampa i campi info dei nodi di albero(r) in ordine prefisso nel modo seguente: salta i primi $n-1$, stampa il nodo n -esimo, poi ne salta $k-1$ stampa quello successivo e così via fino a visitare tutti i nodi) && (restituisce un intero m ($1 \leq m \leq k$) che indica che si dovranno saltare $m-1$ nodi prima di stampare il prossimo)

Per capire la POST, consideriamo l'albero di prima, con $k=4$. Quando `stampa_a_salti` viene invocata sul nodo 4, allora n deve essere 2 e il valore restituito da questa invocazione dovrà essere 1. Il valore restituito 1 indica che si deve stampare il nodo che segue 3 (ultimo nodo dell'albero radicato in 4) nell'ordine prefisso dell'albero e cioè il nodo 2. Invece l'invocazione di `stampa_a_salti` sul nodo 8 avrà $n=1$ (che indica che 8 va stampato) e restituirà 2 che indica che il nodo 3 va saltato per stampare poi il nodo 2. Insomma, per la funzione a cui viene ritornato, l'intero m ha lo stesso scopo del parametro formale n .

Correttezza: dimostrare la correttezza di `stampa_a_salti` rispetto a PRE e POST date.