

Esercizio 1 del 1/6/2020

Si chiede di costruire una libreria di funzioni ricorsive per alberi BST. Le funzioni della libreria sono le seguenti:

- 1) Stampa in formato lineare dell'albero: `void stampa_l(nodo*r)`
- 2) Inserimento di un nodo con `info=x` mantenendo la proprietà BST: `nodo* insert(nodo*r, int x)`
- 3) Ricerca di un nodo con `info=x`: `bool search(nodo*r, int x)`
- 4) Calcolo del campo `info` massimo e minimo : `nodo* max(nodo*r), nodo* min(nodo*r)`
- 5) Calcolo dell'altezza dell'albero : `int altezza(nodo*r)`
- 6) Calcolo della lunghezza minima tra i cammini che collegano la radice ad una foglia:
`int altMin(nodo*r)`

La libreria consiste di un file "BST.h" con la dichiarazione del tipo `nodo` e i prototipi delle funzioni richieste e di un file "BST.cpp" con le implementazioni delle funzioni stesse. Tutte le dichiarazioni in BST.h vanno racchiuse in un namespace che si chiama `BST`. Il file BST.cpp deve includere "BST.h" seguito dall'istruzione "using namespace BST;". Inoltre ogni definizione di funzione in BST.cpp deve specificare che si sta definendo una funzione del namespace `BST` e quindi il nome della funzione deve essere preceduta da "BST::". Per esempio per la funzione `search`, la prima riga deve essere: `bool BST::search(nodo*r, int x)`. Non importa specificare `BST` per le invocazioni ricorsive.

Nel file "BST.cpp" ogni funzione deve essere accompagnata dalla sua PRE e POST e ovviamente le funzioni devono obbedire alle PRE e POST specificate.

Il main "ex-1-main.cpp" deve iniziare con le opportune istruzioni di `include` e `using` e costruisce un piccolo albero BST. Dopo di che deve leggere una sequenza di interi corrispondenti alle operazioni elencate prima. Il comportamento del programma è spiegato nel seguente esempio.

Esempio: supponiamo che l'albero corrente sia `3(1(,), 5(,))` e che si voglia inserire in quest'albero un nodo con `info=4`. L'istruzione che corrisponde a questa richiesta viene rappresentata da: `2 4`. Il valore `2` fa riferimento al punto (2) della precedente lista (che infatti corrisponde alla funzione di inserimento) e `4` è il valore da inserire. Dopo aver eseguito l'inserimento, verrà stampato il nuovo albero (con la funzione di `stampa_l` del punto (1)) che è: `3(1(,), 5(4(,),))`

Il punto (4) corrisponde a 2 funzioni, quindi, se dopo il `4` si legge `1` allora è richiesta la funzione `max`, se si legge `2` è richiesta la funzione `min`. Dopo aver eseguito queste funzioni, va stampato il campo `info` del nodo restituito. Lo stesso vale per le funzioni (5) e (6). Per la funzione (3) di ricerca, si stamperà "valore `x` presente/non presente". In corrispondenza dell'input `1` va semplicemente invocata `stampa_l`.

Per terminare le operazioni il main deve leggere `0`.

Correttezza: scrivere la prova induttiva della funzione (2)

Suggerimenti: Provate a vedere cosa succede se togliete l'operazione di "using namespace BST;" dal main.cpp. Invece di "using namespace BST;" potete invocare ogni funzione specificando `BST::` prima del nome, ma attenzione che `nodo` è definito nel namespace. Provate.