

Esame 20220729

Esercizio Lode

(1) Esercizio Lode

ESSAY marked out of 1 penalty 0 File picker

Scrivere dichiarazione e definizione di una funzione **iterativa** `altezza_iter` che prende come argomento `root` di tipo puntatore a costante ad una struttura `tree` che rappresenta un albero binario di ricerca di interi (si veda il codice in `lode.cpp`).

La funzione calcola in modo **iterativo** l'altezza dell'albero preso come argomento. La definizione (ricorsiva) di altezza di un albero è così definita:

```
int altezza_ric(const tree * root) {
    if (NULL == root) return 0;
    else return 1 + std::max(altezza_ric(root->left),
                             altezza_ric(root->right));
}
```

L'implementazione della funzione `altezza_iter` se internamente usa strutture dati di supporto che prevedono allocazione di memoria, deve preoccuparsi di deallocare la memoria allocata per evitare *memory leak*.

Qui di seguito è riportato un frammento dell'esecuzione del programma:

```
Initial tree:
< omesso per questioni di spazio >
Altezza ricorsiva = 13
Altezza iterativa = 13
```

Note:

- Scaricare il file `lode.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `altezza_iter`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.
- **Suggerimento:** effettuare un attraversamento dell'albero per livelli. Calcolo numero totale di nodi al livello corrente, ed elaboro ogni nodo al livello corrente e se i figli sono diversi da `NULL` ne tengo traccia per processarli dopo (e.g. inserisco in una coda). Incremento un counter ad ogni livello. Quindi l'altezza di un albero é uguale al numero totale di livelli in esso contenuti. Per comodità se ritenuto utile è fornita l'implementazione di una coda di costanti puntatori a `tree` che mette a disposizione metodi: `init`, `deinit`, `first`, `enqueue`, `dequeue`, `size` (si veda il file `lode.cpp` per i dettagli).
- Le uniche assunzioni che si possono fare sul contenuto dell'albero binario di ricerca e sul numero di elementi salvati nell'albero binario di ricerca sono **solo quelle espressamente specificate in questo testo** (e NON quelle riportate nel file fornito, che sono SOLO indicative per consentire di svolgere l'esame).
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `cstddef`, `cstdlib`, o `ctime`.

Information for graders:

Total of marks: 1