

Laboratorio di Algoritmi e Strutture Dati

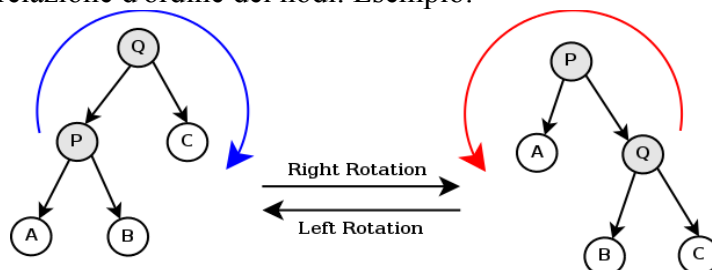
a.a. 2016-2017

Prof. Murano Aniello - Dott. Di Stasio Antonio

Progetto n°2

1 – Traccia

1. Sia T un albero binario di ricerca. Si modifichino le funzioni di gestione di un ABR in modo da poter gestire l'inserimento di chiavi uguali in T , adottando la strategia di soluzione più efficiente.
 - Quali sono le prestazioni asintotiche delle funzioni modificate? (**Giustificare la risposta**)
2. Le operazioni di inserimento e cancellazione su T hanno costo $O(h)$, dove h è l'altezza dell'albero. Si implementi una procedura che crei in modo casuale un albero binario di ricerca. Determinare quindi l'altezza media di un albero di ricerca creato casualmente. Per far ciò si osservi l'altezza di una serie di alberi binari di ricerca (il cui numero non è fissato a priori) di uguale dimensione creati in modo casuale e si determini quale funzione matematica descrive l'altezza media degli alberi creati.
3. Implementare la funzione MERGE che dati due ABR $T1$ e $T2$ li unisca in un solo albero senza utilizzare un terzo albero di appoggio. Qual è la complessità della funzione MERGE (**Giustificare la risposta**).
4. Implementare la funzione ROTATION che dato in input un ABR, una direzione (left o right) ed un intero n , esegua n volte la rotazione dell'albero nella direzione indicata, senza modificare la relazione d'ordine dei nodi. Esempio:



5. Attraverso la funzione ROTATION trasformare un albero non bilanciato in uno bilanciato.
6. (Opzionale) Creare una funzione generica che stampi graficamente l'albero a video.

2 – Consegna

La consegna deve essere effettuata tramite invio per posta elettronica di un unico file .zip.

All'interno di questo archivio compresso devono trovare posto:

- 1) Cartella contenente il codice sorgente completo ;
- 2) Non è obbligatorio sviluppare il progetto con l'utilizzo di IDE quali dev-c++/code blocks etc. Anzi è **FORTEMENTE CONSIGLIATO** sviluppare il progetto senza utilizzo di IDE. In tal caso bisogna **OBBLIGATORIAMENTE** scrivere nella relazione la riga di comando da lanciare da terminale. Es: gcc main.c function1.c function2.c etc. In caso in cui utilizzate IDE inviare **ANCHE** l'eseguibile dell'applicazione.
- 3) una relazione che descriva il vostro lavoro sviluppata secondo quanto descritto nel file 'come strutturare la relazione'.

Il file .zip deve avere un nome il cui formato è **OBBLIGATORIO**: pr2grxx.zip, dove, ovviamente xx sta per il numero del gruppo. Richiamare questa struttura di nomi anche per il file della relazione rel1grxx.pdf. Non ci sono restrizioni al nome dei file del codice sorgente.

La deadline è il 12/05/17 alle ore 23:59. La mail con la consegna deve essere inviata a antonio.distasio@unina.it, il subject deve contenere OBBILAGORIAMENTE la seguente: 'LASD consegna pr1grxx'.

Nel caso di più consegne prenderemo in considerazione solo la più **RECENTE**. Cercate di mettervi d'accordo su chi manda la versione finale ed evitate di spedire le consegne da indirizzi differenti ogni volta.