Marco Carega-20018128 Tommaso Pessina-20018259 Stefano Panelli-20020197

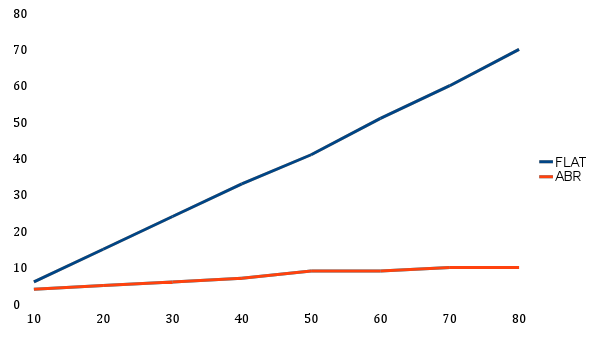
Esercitazione n. 3: Alberi Binari di Ricerca (ABR)

E' richiesta la creazione di un programma che modelli la complessità media della ricerca dati su ABR confrontandola alla ricerca dati in archivio flat, visualizzando questa complessità come numero di nodi visitati. Per fare ciò il programma deve caricare nell'ABR un numero di studenti N deciso dall'utente ed eseguire due cicli di ricerche randomizzate; uno utilizzando l'ABR ed uno utilizzando questo come se fosse un archivio “flat” calcolando la complessità di ciascuno.

Il metodo usato per definire la ricerca casuale consiste nella ricerca di un dato indicizzato da un numero scelto in maniera randomica tra le chiavi utilizzate.

Analizzando il programma creato, dopo le varie funzioni di creazione dell'Albero si può notare che è stata scelta la versione ricorsiva dell'implementazione per quanto riguarda la ricerca avvalendosi delle proprietà dell'ABR.

Per quanto riguarda invece la ricerca dati in archivio flat è stato utilizzato l'algoritmo di ricerca in post-ordine, questo opera effettuando prima le due chiamate (sinistra e destra) e poi eseguendo le operazione sul nodo corrente.

Dopo che il programma ha scritto i risultati in un file di testo, si passa a creare un grafico per confrontarli: in esso, l'asse x rappresenta il numero N di studenti (preso per comodità da 10 a 80) mentre l'asse y rappresenta il numero di HIT (complessità media).

Da quello che si è potuto vedere con questo test possiamo dire che il sistema ad ABR è senza dubbio conveniente rispetto alla controparte a ricerca flat nonostante la più grande difficoltà di implementazione, in quanto, non solo una ricerca tramite ABR ha un numero di HIT molto più basso ma esso rimane più “costante” all'aumento del numero N.

Inoltre si può notare che il grafico corrisponde all'andamento della complessità teorica (grazie anche al fatto che è stata raccolta una quantità considerevole di dati per garantire un risultato omogeneo). Bisogna comunque tenere in conto che in caso di alberi non bilanciati la complessità pratica non corrisponde a quella teorica in quanto, in questo caso, quest'ultima tenderebbe a N e non a Log(n) (come ci si aspetterebbe dai dati teorici); cosa che non succede nel caso degli AVL in quanto in caso di sbilanciamento l'albero verrebbe ribilanciato, per cui l'uso di questi è una ottimizzazione da considerarsi nei casi in cui grandi sbilanciamenti “costringono” all'uso di algoritmi a ricerca flat.

Per quanto riguarda invece la possibilità di usare una cosiddetta struttura d’indicizzazione, si può dire che, grazie all’indicizzazione, i tempi di ricerca sono più efficienti rispetto a un albero senza questa implementazione. Nel caso non si potesse utilizzare la ricerca indicizzata, si è costretti a scansionare l’albero totalmente utilizzando altri metodi di ricerca (in questo caso, per nome o per cognome). Per risolvere il problema, si può indicizzare un ABR anche per nome/cognome.