Albero Binario di ricerca vs Albero Rosso Nero

Indice

1 Introduzione

Lo scopo dell'elaborato consiste nel mettere a confronto due tipi di alberi binario : albero binario di ricerca e albero rosso nero. Entrambi permettono di effettuare in maniera efficiente operazioni come: ricerca, inserimento e cancellazione di elementi.

2 Descrizione strutture

In informatica un albero è la struttura dati che si riconduce al concetto di albero con radice presente nella teoria dei grafi. Un albero si compone di due tipi di sotto strutture fondamentali: il nodo, che in genere contiene informazioni e l'arco che stabilisce un collegamento gerarchico tra due nodi: si parla allora di un nodo padre dal quale esce un arco orientato che lo collega ad un nodo figlio. Un albero può essere classificato in ordinato e non ordinato. I primi non seguono alcuna regola per quanto riguarda le relazioni padre-figlio, mentre i secondi impongono che tra il nodo padre e i nodi figli ci sia un ben preciso ordinamento.
Un'altra classificazione può essere fatta in base al numero massimo di figli che un nodo può avere. Si può parlare dunque di alberi binari in cui ogni nodo può avere al massimo due figli.

2.1 Albero binario di ricerca ABR

Un albero binario supporta molte operazioni dinamiche e ha le seguenti proprietà:

1. Il sotto-albero sinistro di un nodo x contiene soltanto i nodi con chiavi minori della chiave del nodo x.
2. Il sotto-albero destro di un nodo x contiene soltanto i nodi con chiavi maggiore della chiave del nodo x.
3. Il sotto-albero destro e il sotto-albero sinistro di un nodo x devono essere entrambi due alberi binari di ricerca.

2.2 Albero rosso nero RBT

Un albero rosso nero è un particolare albero binario di ricerca, in cui l'altezza (grazia alle sue proprietà) rimane limitata. Questo implica delle operazioni di inserimento ed eliminazione più complesse rispetto a quelle di semplici alberi binari di, ma garantiscono che esse vengano eseguite in $O(logn)$. In particolare sono ottimi nel caso peggiore, cioè quando si ha un albero completamente sbilanciato.

- ogni nodo x ha un attributo booleano x.color

- x.color: rosso o nero
- Eredita gli attributi dell'albero binario di ricerca.
- Foglie vuote (NIL e nere).

2.2.1 Proprietà RBT

1. Ogni nodo è rosso o nero.
2. La radice è nera
3. Ogni foglia (T.NIL) è nera.
4. Se un nodo è rosso, allora entrambi i suoi figlio sono neri (No due rossi consecutivi in un cammino semplice da radice a foglia)
5. Tutti i cammini da ogni nodo alle foglie contengono lo stesso numero di nodi neri.

3 Prestazioni attese

Le operazioni di base su un albero binario di ricerca richiedono tempo proporzionale all'altezza dell'albero, quindi se n è il numero di nodi:

- $\Theta(logn)$ nel caso migliore (albero completo)
- $\Theta(n)$ nel caso peggiore (albero completamente sbilanciato
- $\Theta(logn)$ nel caso medio

Invece, come descritto precedentemente l'albero rosso-nero ha migliore prestazioni nel caso peggiore rispetto all'albero binario di ricerca.

4 Esperimenti

Ho effettuato esperimenti mirati ad evidenziare la migliore performance dell'albero rosso nero nel caso medio. Dopo aver creato i due alberi sbilanciati con gli stessi elementi vengono fate delle prove ripetute, calcolandone i tempi di esecuzione, sulla ricerca di un elemento al loro interno.

4.1 Grafici

I grafici riportati, mettono a confronto le due strutture nel caso peggiore e nel caso medio.

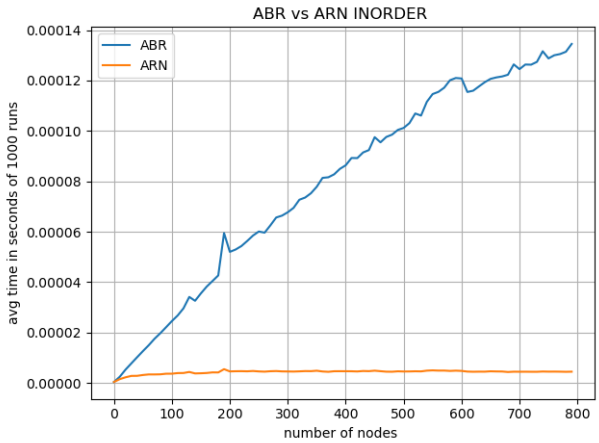

Figura 1: Tempo per la ricerca di un elemento all'interno di ABR e ARN al crescere dei numero dei nodi nel caso peggiore

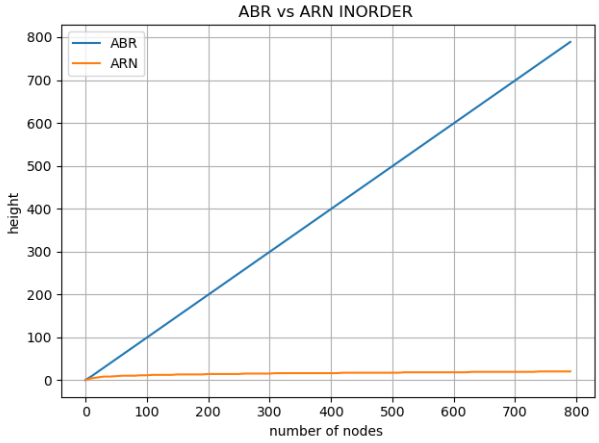

Figura 2: Altezza per ABR e ARN al crescere dei numero dei nodi nel caso peggiore

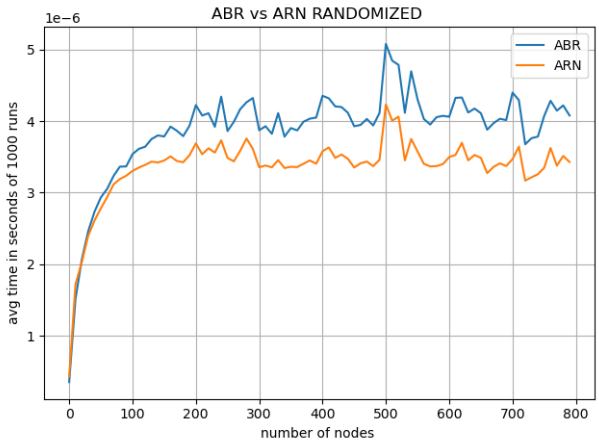

Figura 3: Tempo per la ricerca di un elemento per ABR e ARN al crescere dei numero dei nodi nel caso medio

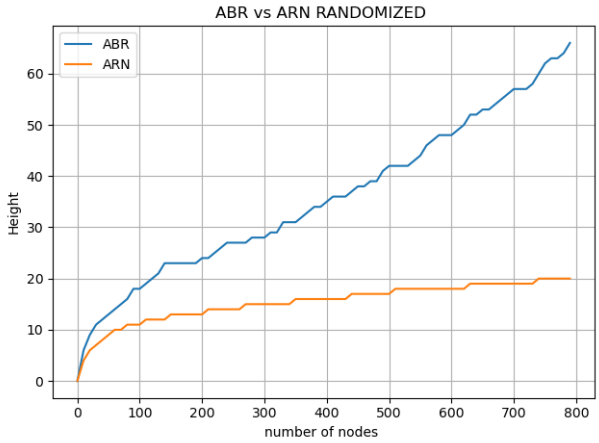

Figura 4: Altezza per ABR e ARN al crescere dei numero dei nodi nel caso medio

5 Documentazione del codice

L'elaborato scritto in linguaggio Python oltre ad importare le librerie per la generazione di numeri pseudo casuali e generazione di grafici, contiene due classi che implementano rispettivamente l'albero binario di ricerca e albero rosso nero Le due classi contengono i metodi di inserimento, cancellazione e ricerca .

6 Conclusione

Gli esperimenti evidenziano che nel caso medio entrambe le strutture hanno un andamento logaritmico, mentre nel caso peggiore l'albero binario di ricerca ha un andamento lineare e l'albero rosso nero ha un andamento logaritmico che è migliore.

7 Informazioni Sul Calcolatore

7.1 Hardware

CPU: Intel Core i7-8565u @ 1.80Ghz
RAM: 8 GB DDR3
Graphic: Intel HD Graphics 620

7.2 Software

OS: Windows 10 Home
IDE: Pycharm Community Edition 2020.1.1