Informatica - Area scientifica
Dipartimento di Scienze matematiche, informatiche e multimediali
Università di Udine

Seconda parte del progetto di ASD

Enrico Martin martin.enrico@spes.uniud.it 145175
Luca Bazzetto bazzetto.luca@spes.uniud.it 144760
Andrea Bordignon bordignon.andrea001@spes.uniud.it 142295

Anno accademico 2020/2021

Indice

Capitolo 1

Consegna

La seconda parte del progetto richiede l'implementazione e l'analisi dei tempi di l'esecuzione di operazioni di inserimento e di ricerca in alberi binari di ricerca. Nello specifico, si richiede di implementare le operazioni di ricerca e inserimento per tre tipi diversi di alberi binari di ricerca: alberi binari di ricerca semplici, di tipo AVL e di tipo Red-Black.
Si assuma che ogni nodo di un albero binario di ricerca contenga una chiave numerica (di tipo intero) e un valore alfanumerico (di tipo stringa). Non è richiesta l'implementazione dell'operazione di rimozione di un nodo (i test automatici di auto-valutazione sono preparati in modo da non eseguire mai l'istruzione di rimozione).

Si richiede una stima dei tempi medi e ammortizzati per l'esecuzione di n operazioni di inserimento e ricerca nei tre tipi di alberi binari di ricerca sopra descritti.
Per tale stima si potrà procedere nel modo seguente:
Al variare del parametro n, ad esempio, fra 1000 e 1000000, si eseguono n volte le seguenti operazioni su un albero di ricerca inizialmente vuoto: si genera in modo pseudo-casuale un valore intero k, si ricerca un nodo con chiave k nell'albero e, qualora il nodo non esistesse, si inserisce un nuovo nodo con chiave k nell'albero. Si noti che al termine di tale procedura saranno state eseguite esattamente n operazioni di ricerca e m operazioni di inserimento, per un opportuno $m \leq n$ (l'albero binario di ricerca conterrà quindi al più n nodi). Nell'ipotesi che i numeri generati in modo pseudo-casuale varino in un dominio sufficientemente grande, il valore di m sarà probabilmente simile a quello di n e potrà quindi essere sostituito con n.

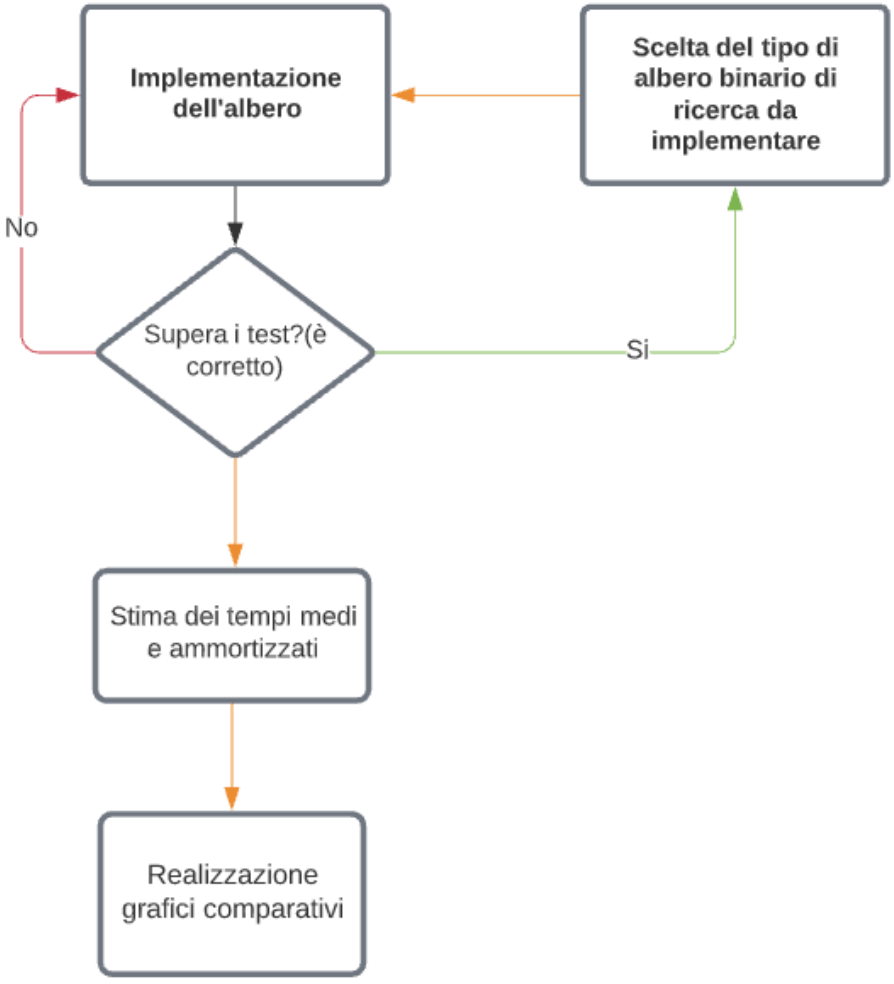
Si richiede di stimare il tempo ammortizzato al variare di n con un errore relativo limitato superiormente da 0.01.

Capitolo 2

Approccio al progetto

ll primo approccio al progetto è stato quello di eseguire un'attenta lettura della consegna al fine di evidenziare in primis le varie richieste e contemporaneamente di effettuare una valutazione preliminare sugli obiettivi che sarebbero potuti risultare più difficili da implementare.

Successivamente è stato realizzato un diagramma, qui di seguito riportato, che sintetizza la metodologia globale con cui approcciarsi alla realizzazione del progetto.
Il progetto infine è stato verosimilmente così svolto.
Abbiamo, infatti, prima di tutto implementato i BST richiesti, successivamente testato la loro correttezza e una volta completati abbiamo poi calcolato la stima dei tempi medi e infine realizzato i grafici comparativi con le dovute scale.



Capitolo 3

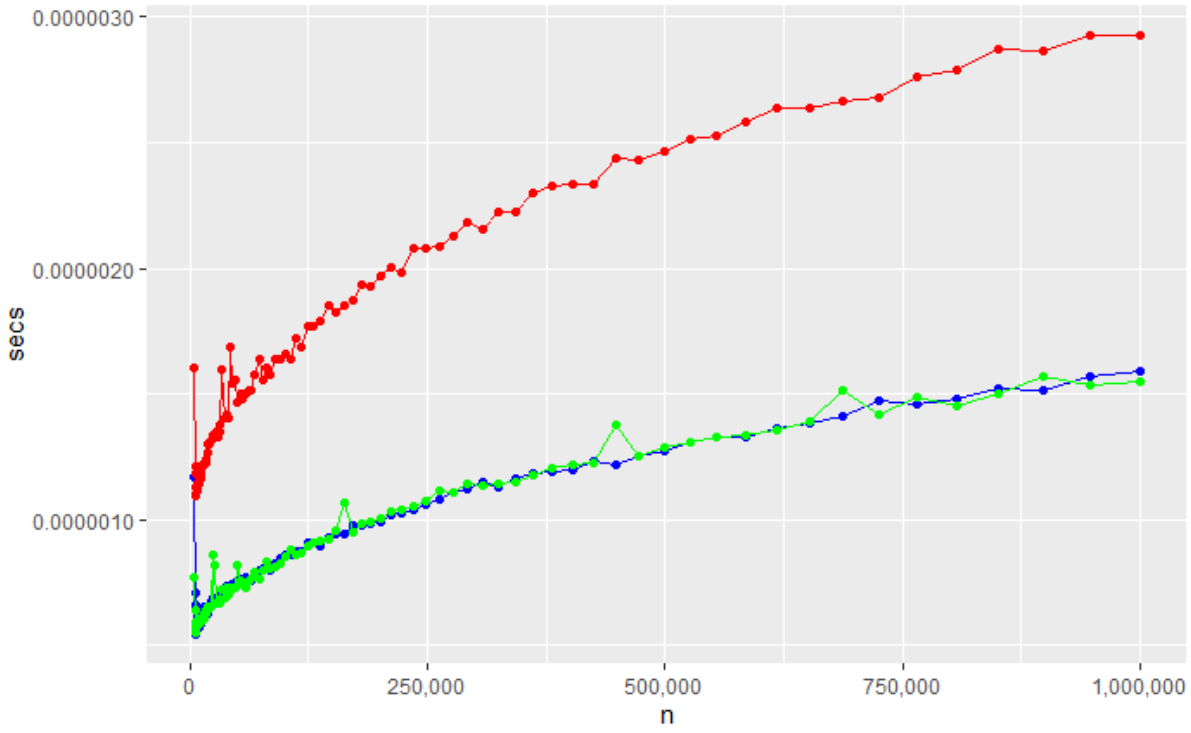
Analisi dei Tempi

In questa sezione della relazione si riporteranno i grafici delle misurazioni dei tempi.
I tempi sono stati misurati per ogni albero binario in 100 differenti esecuzioni.
I grafici mettono sempre in relazione i tre alberi binari di ricerca distinti dai seguenti colori:

- Rosso: AVL
- Blu: BST
- Verde: RBT

3.1 Grafico 1

Il primo grafico mostra l'andamento logaritmico crescente in funzione della dimensione dell'input dei tre algoritmi in funzione di n.
Nell'asse delle ascissa troviamo n -la dimensione dell'input- mentre nell'ordinata troviamo il tempo espresso in secondi.
Si notano poi diversi valori dall'andamento altalenante -non costante- sopratutto nella fase iniziale, traducibili in termini di costi di overhead.



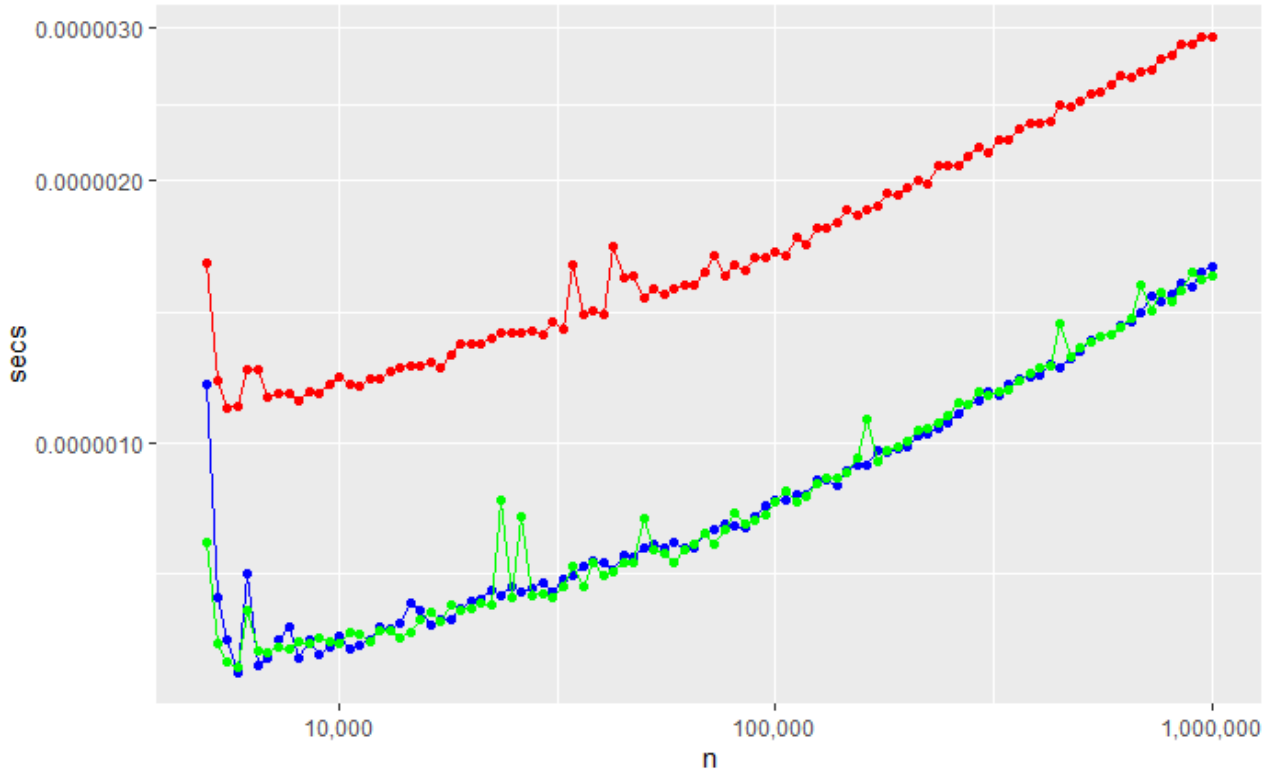
3.2 Grafico 2

In questo secondo grafico è stata applicata una scala logaritmica per ogni asse. Questo fa risultare i campioni distribuiti in maniera uniforme, tuttavia rimane coerente con il grafico precedente e ci permette di capire quale dei tre algoritmi sia il più efficiente e, ovviamente, quale il meno.

La lettura del grafico infatti ci permette di ordinare gli algoritmi in termini di efficienza (dal più vantaggioso al meno) e stilarne la seguente "classifica":

- 1. BST
- 2. RBT
- 3. AVL

Si osserva anche che gli algoritmi BST e RBT differiscono di poco e non è immediato riconoscere quale dei due sia più efficiente rispetto all'altro.
Immediato è invece il distacco con AVL.



Capitolo 4

Conclusioni

I tre algoritmi, come già riportato, presentano un andamento logaritmico. Essendo nota la teoria dei seguenti alberi binari di ricerca ritroviamo una certa coerenza -in termini di implementazione e di misurazione dei tempi- nel momento in cui il loro andamento rispecchia la loro complessità asintotica nel caso medio, ovvero: $O(log(n))$.
Riportiamo inoltre i calcoli della media e della deviazione standard per ogni algoritmo:

- BST
  - **media:** 0.0000894534
  - **deviazione standard:** 0.0000000001
- AVL
  - **media:** 0.0001737295
  - **deviazione standard:** 0.0000000003
- RBT
  - **media:** 0.0000894632
  - **deviazione standard:** 0.0000000001

In conclusione decretiamo l'algoritmo BST come più vantaggioso , d'altra parte è l'algoritmo AVL a rivelarsi come meno vantaggioso.
Inoltre i risultati della deviazione standard per i due algoritmi più efficienti ci danno un riscontro di un uniformità di valori migliore rispetto al restante algoritmo.