

Analisi delle prestazioni di alberi di ricerca

Alberi binari di ricerca e Alberi rosso-neri a confronto

Leonardo Gori

January 24, 2021

Abstract

Alberi binari di ricerca (Binary Search Trees) e alberi rosso-neri (Red-Black Trees) sono strutture dati che rappresentano un insieme di unità dette 'nodi', collegati tra loro da archi secondo regole diverse. Essi hanno vari ambiti applicativi nel campo dell'informatica in quanto permettono in modo efficiente l'inserimento, la rimozione e la ricerca di elementi di un insieme. In questa relazione verranno analizzate e comparate le prestazioni temporali per le operazioni di ricerca e inserimento di entrambe le strutture dati.

1 Introduzione

Alberi binari di ricerca (ABR) e alberi rosso-neri (ARN) sono costituiti da nodi collegati da archi secondo una relazione d'ordine predefinita. Entrambe le strutture dati sono accomunate dal fatto di avere una struttura a albero binario, che ha origine da un nodo con puntatore al padre nullo, detto 'radice', e che terminano con nodi con puntatori ai figli nulli, detti 'foglie'.

1.1 Alberi binari

Un albero binario è un grafo non diretto, connesso e aciclico i quali nodi sono collegati da un numero di archi compreso tra 0 e 2. Ogni albero binario è costituito da un nodo di partenza (detto 'radice') dal quale si diramano al massimo due sotto alberi binari, fino a concludersi con nodi senza successori (detti 'foglie').

1.2 Alberi binari di ricerca

Un ABR estende la struttura dell'albero binario, è costituito da nodi ai quali vengono associati:

- puntatore al padre
- chiave
- puntatore al figlio destro
- puntatore al figlio sinistro

La relazione d'ordine degli ABR è basata sul valore della chiave di ogni nodo; presi due nodi di un albero di ricerca generico x e y :

- Se il nodo y è nel sottoalbero sinistro di x , allora $y.key \leq x.key$;
- Se il nodo y è nel sottoalbero destro di x , allora $y.key \geq x.key$;
- Il sottoalbero destro e il sottoalbero sinistro devono essere entrambi due alberi binari di ricerca.

1.3 Alberi rosso-neri

Un ARN è un'entità software che estende la struttura dell'albero binario di ricerca associando ai nodi che lo compongono una proprietà aggiuntiva, il colore, che può essere come si evince dal nome che lo identifica, rosso oppure nero. Inoltre a ogni nodo non è concesso di avere un puntatore nullo, di default il puntatore è assegnato a nodi 'foglia' con attributi nulli (T.nil), ovvero che non contengono dati. Oltre a ereditare le proprietà degli alberi binari di ricerca, gli ARN rispettano le seguenti proprietà:

- Ogni nodo è rosso o nero
- La radice è nera
- Ogni foglia (T.nil) è nera
- Se un nodo è rosso, allora entrambi i suoi figli sono neri (non esistono quindi due nodi rossi consecutivi in un cammino semplice da radice a foglia)
- Tutti i cammini da ogni nodo alle foglie contengono lo stesso numero di nodi neri

2 Obiettivi dell'elaborato

Obiettivo principe di questo elaborato è quello di mostrare le principali differenze delle due strutture dati, in particolare mostrando come gli alberi rosso-neri aumentino le prestazioni degli alberi binari di ricerca in termini di tempo e di calcolo computazionale. Come vedremo nel corso della relazione, gli ARN hanno la peculiarità di essere alberi binari di ricerca in grado di bilanciare automaticamente la propria struttura, mantenendo quindi una sorta di simmetria che permette di ottimizzare operazioni di ricerca al loro interno.

2.1 ABR e ARN: Teoria e casi particolari

Secondo la teoria i tempi di esecuzione di operazioni di attraversamento, ricerca e inserimento per gli ABR sono linearmente dipendenti dall'altezza h della struttura stessa dell'albero, dove per altezza si intende la massima distanza di una foglia dalla radice. L'altezza di un ABR è condizionata dall'ordine di inserimento delle chiavi dei nodi:

- l'inserimento di chiavi ordinate in modo totalmente o parzialmente crescente/decescente rappresenta il caso peggiore, poichè la struttura del ABR apparirà 'sbilanciata' rispettivamente verso destra/sinistra, quindi l'altezza dell'albero risulterà essere uguale o di poco minore al numero di nodi ($h \simeq n$);
- L'inserimento di chiavi in modo casuale rappresenta quindi il caso migliore in quanto la struttura dell'ABR sarà bilanciata e l'altezza dell'albero seguirà una legge logaritmica (in base 2) dipendente dal numero dei nodi ($h \simeq \log_2 n$).

Come abbiamo accennato precedentemente gli ARN hanno la proprietà di essere strutture dati auto bilancianti, la quale permette loro di mantenere un'altezza logaritmica per qualsiasi ordine di inserimento di chiavi col costo di algoritmi di inserimento più complessi.

Ordine di inserimento	Attraversamento, ricerca e inserimento ABR	Attraversamento, ricerca e inserimento ARN
Crescente/Decrescente	$O(n)$	$O(\log_2 n)$
Riempito in modo casuale	$O(\log_2 n)$	$O(\log_2 n)$

3 Test delle prestazioni

Il progetto allegato alla relazione si occupa di testare le prestazioni di inserimento e di ricerca delle due strutture dati, mostrando su grafici i tempi medi di completamento delle operazioni in relazione al numero dei nodi degli alberi. In particolare vengono presentati alberi ABR e ARN di dimensioni arbitrarie costruiti su ordine di inserimento crescenti e casuali (è stato deciso di omettere il caso di ordinamento decrescente poichè analogo al caso crescente). Ogni albero viene creato inserendo dinamicamente i nodi, registrando il tempo di inserimento totale per la sua costruzione completa. Fissata la dimensione, questo procedimento viene eseguito per un numero k arbitrario di volte (di default 100) e successivamente calcolata la media dei tempi. Stessa cosa vale per l'operazione di ricerca: fissata la dimensione, per k volte, ogni albero generato viene interrogato della totalità dei suoi nodi, registrandone i tempi e calcolando successivamente la media.

3.1 Aspettative

Come già visto precedentemente ci si aspetta che per i casi di inserimento ordinati le prestazioni degli ABR siano migliori, mentre per casi di inserimento casuale le prestazioni siano simili o linearmente dipendenti.

4 Risultati degli esperimenti

Di seguito vengono presentati tabelle e grafici che mostrano i risultati degli esperimenti condotti su alberi di dimensioni variabili tra 20 e 500 nodi di passo 20.

4.1 Ordine di inserimento crescente

4.1.1 Tempi di inserimento

Dimensione	Inserimento ordinato ARN	Altezza ARN	Inserimento ordinato ABR	Altezza ABR
20	2.660e-04	5	1.478e-04	19
40	5.308e-04	7	5.366e-04	39
60	8.283e-04	8	1.159e-03	59
80	1.245e-03	9	2.057e-03	79
100	1.563e-03	10	3.164e-03	99
120	1.970e-03	10	4.530e-03	119
140	2.328e-03	11	6.209e-03	139
160	2.697e-03	11	7.979e-03	159
180	3.154e-03	11	1.007e-02	179
200	3.567e-03	12	1.232e-02	199
220	3.974e-03	12	1.505e-02	219
240	4.416e-03	12	1.796e-02	239
260	4.872e-03	13	2.134e-02	259
280	5.313e-03	13	2.461e-02	279
300	5.780e-03	13	2.834e-02	299
320	6.162e-03	13	3.194e-02	319
340	6.734e-03	13	3.606e-02	339
360	7.393e-03	13	4.056e-02	359
380	7.444e-03	13	4.525e-02	379
400	7.950e-03	14	5.040e-02	399
420	8.334e-03	14	5.545e-02	419
440	8.771e-03	14	6.078e-02	439
460	9.640e-03	14	6.625e-02	459
480	9.783e-03	14	7.267e-02	479
500	1.027e-02	14	7.850e-02	499

Table 1: Tempo medio impiegato per inserire valori in ordine crescente all'interno di ABR e ARN e relative altezze medie

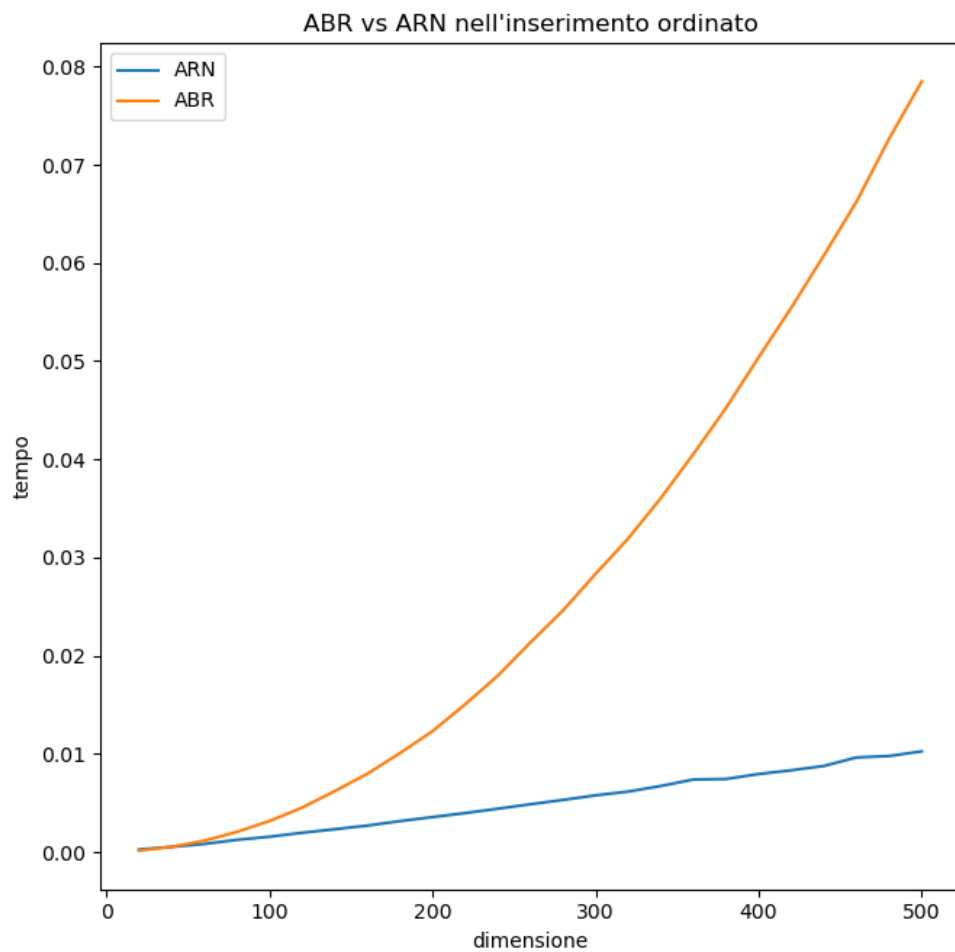


Figure 1: Grafico che descrive i tempi medi di inserimento ordinato in relazione al numero di nodi.

4.1.2 Tempi di ricerca

Dimensione	Ricerca in ARN	Altezza ARN	Ricerca in ABR	Altezza ABR
20	3.729e-05	5	1.040e-04	19
40	9.142e-05	7	4.077e-04	39
60	1.672e-04	8	9.069e-04	59
80	2.215e-04	9	1.533e-03	79
100	3.096e-04	10	2.463e-03	99
120	3.738e-04	10	3.495e-03	119
140	4.585e-04	11	4.792e-03	139
160	5.305e-04	11	6.266e-03	159
180	6.179e-04	11	7.987e-03	179
200	7e-04	12	9.878e-03	199
220	8.100e-04	12	1.196e-02	219
240	8.557e-04	12	1.416e-02	239
260	9.740e-04	13	1.666e-02	259
280	1.061e-03	13	1.924e-02	279
300	1.180e-03	13	2.223e-02	299
320	1.213e-03	13	2.531e-02	319
340	1.325e-03	13	2.871e-02	339
360	1.421e-03	13	3.226e-02	359
380	1.526e-03	13	3.592e-02	379
400	1.599e-03	14	3.983e-02	399
420	1.681e-03	14	4.402e-02	419
440	1.822e-03	14	4.854e-02	439
460	1.917e-03	14	5.300e-02	459
480	1.971e-03	14	5.779e-02	479
500	2.079e-03	14	6.262e-02	499

Table 2: Tempo medio impiegato per la ricerca di valori inseriti in ordine crescente all'interno di ABR e ARN e relative altezze medie

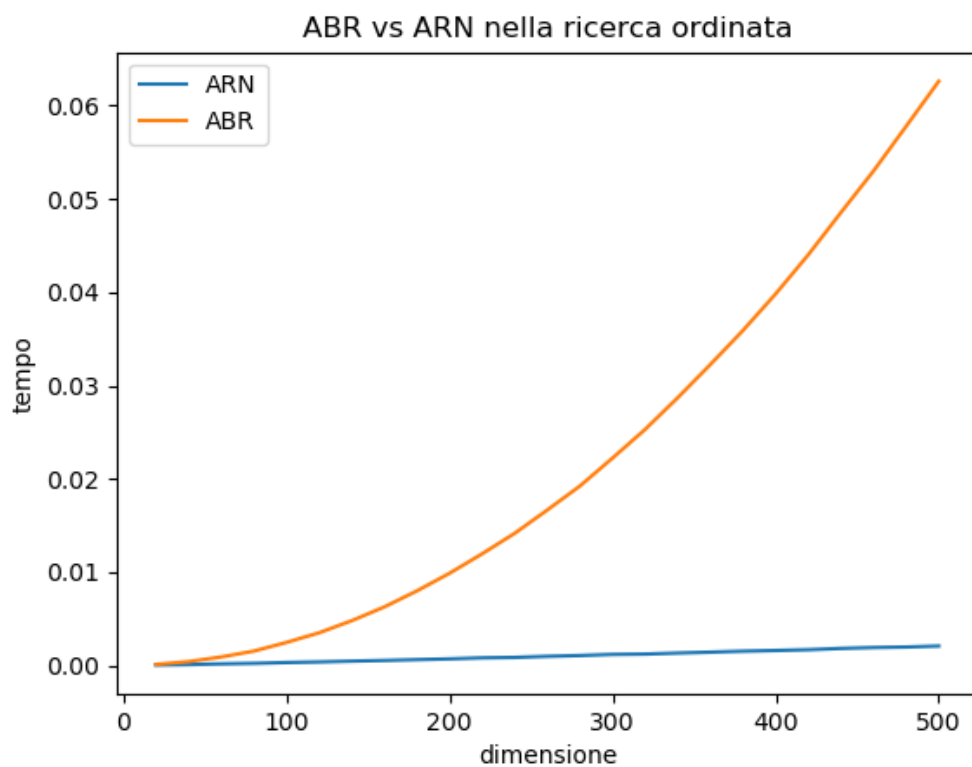


Figure 2: Grafico che descrive i tempi medi di ricerca per valori inseriti in ordine crescente in relazione al numero di nodi.

4.2 Inserimento casuale

4.2.1 Tempi di inserimento

Dimensione	Inserimento in ARN	Altezza ARN	Inserimento in ABR	Altezza ABR
20	1.824e-04	5.360	6.565e-05	6.790
40	4.750e-04	8.330	1.495e-04	9.030
60	6.404e-04	9.870	2.629e-04	10.370
80	9.719e-04	11.440	3.708e-04	11.590
100	1.205e-03	12.290	5.005e-04	12.270
120	1.537e-03	13.540	6.140e-04	12.990
140	1.834e-03	14.650	7.693e-04	13.570
160	2.171e-03	15.640	9.131e-04	13.990
180	2.483e-03	16.230	1.026e-03	14.720
200	2.897e-03	16.250	1.130e-03	14.960
220	3.179e-03	17.390	1.352e-03	15.440
240	3.443e-03	18.770	1.454e-03	15.330
260	3.731e-03	18.640	1.687e-03	15.780
280	4.098e-03	18.860	1.800e-03	15.910
300	4.553e-03	19.760	1.829e-03	16.400
320	4.941e-03	20.190	2.009e-03	16.620
340	5.187e-03	21.640	2.336e-03	16.540
360	5.644e-03	21.140	2.276e-03	17.030
380	6.081e-03	21.640	2.464e-03	17.280
400	6.414e-03	21.650	2.610e-03	17.500
420	6.665e-03	22.270	2.788e-03	17.550
440	7.010e-03	22.810	2.886e-03	17.850
460	7.527e-03	22.890	3.004e-03	17.650
480	7.723e-03	23.260	3.326e-03	18.180
500	8.191e-03	23.750	3.538e-03	18.030

Table 3: Tempo medio impiegato per inserire valori casualmente all'interno di ABR e ARN e relative altezze medie

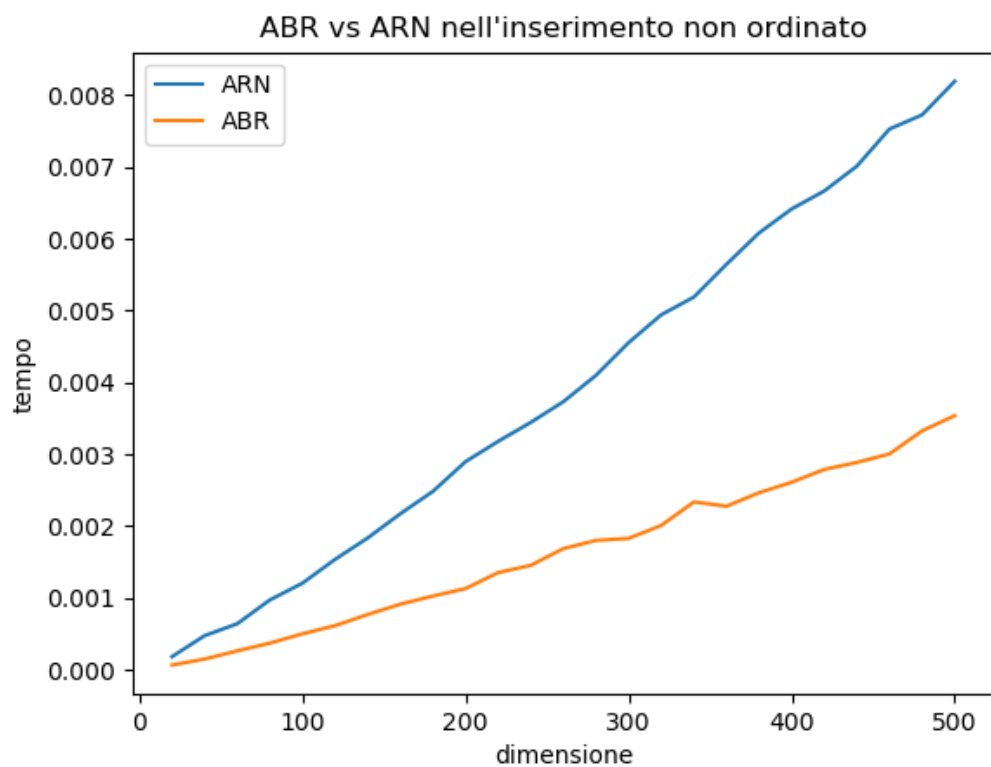


Figure 3: Grafico che descrive i tempi medi di inserimento casuale in relazione al numero di nodi.

4.2.2 Tempi di ricerca

Dimensione	Ricerca in ARN	Altezza ARN	Ricerca in ABR	Altezza ABR
20	4.082e-05	5.360	4.446e-05	6.790
40	1.104e-04	8.330	1.132e-04	9.030
60	1.934e-04	9.870	1.956e-04	10.370
80	2.686e-04	11.440	2.728e-04	11.590
100	4e-04	12.290	3.707e-04	12.270
120	4.872e-04	13.540	4.724e-04	12.990
140	6.188e-04	14.650	5.756e-04	13.570
160	6.881e-04	15.640	6.588e-04	13.990
180	8.357e-04	16.230	7.672e-04	14.720
200	9.371e-04	16.250	8.857e-04	14.960
220	1.075e-03	17.390	9.851e-04	15.440
240	1.217e-03	18.770	1.063e-03	15.330
260	1.357e-03	18.640	1.226e-03	15.780
280	1.527e-03	18.860	1.303e-03	15.910
300	1.626e-03	19.760	1.408e-03	16.400
320	1.767e-03	20.190	1.545e-03	16.620
340	2.058e-03	21.640	1.626e-03	16.540
360	2.007e-03	21.140	1.854e-03	17.030
380	2.247e-03	21.640	1.906e-03	17.280
400	2.334e-03	21.650	2.004e-03	17.500
420	2.479e-03	22.270	2.138e-03	17.550
440	2.735e-03	22.810	2.245e-03	17.850
460	2.791e-03	22.890	2.361e-03	17.650
480	2.941e-03	23.260	2.581e-03	18.180
500	3.115e-03	23.750	2.604e-03	18.030

Table 4: Tempo medio impiegato per la ricerca di valori inseriti casualmente all'interno di ABR e ARN e relative altezze medie

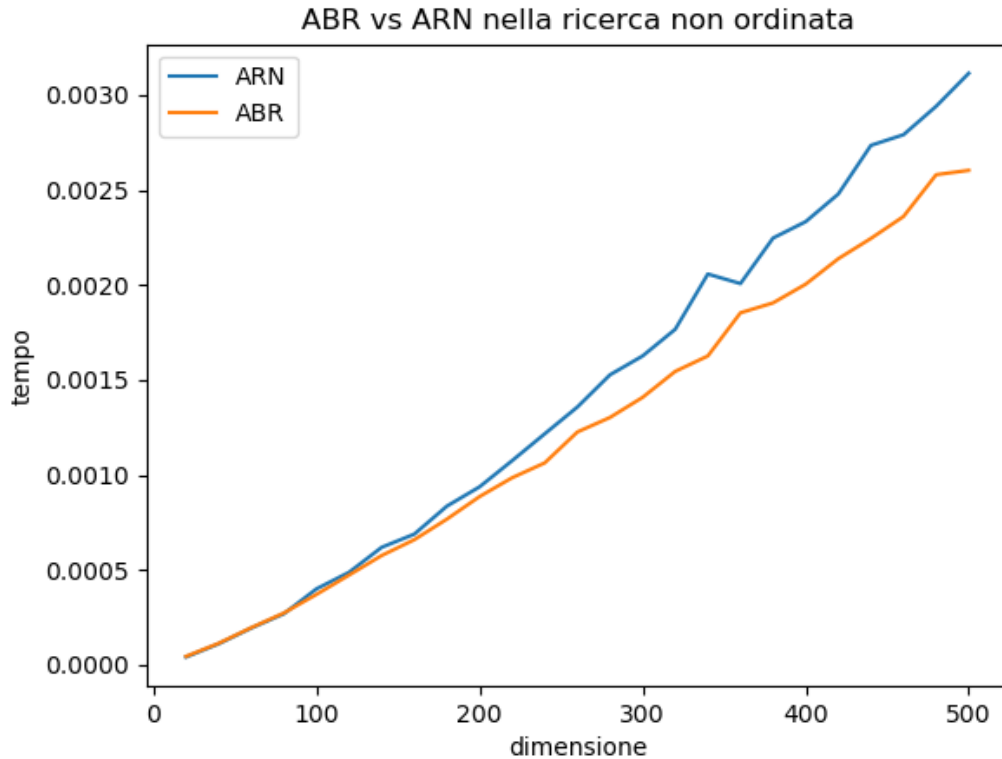


Figure 4: Grafico che descrive i tempi medi di ricerca per valori inseriti in modo casuale in relazione al numero di nodi.

5 Conclusioni

Come si può vedere dai grafici i test confermano la teoria. Infatti per alberi generati su ordine di inserimento crescente le operazioni di ricerca e inserimento sono in media molto più performanti per gli ARN. Per quanto riguarda gli alberi generati su ordine di inserimento casuale, gli ABR sono in genere di poco migliori degli ARN per l'operazione di inserimento; ciò può essere giustificato dal fatto che essendo entrambi generati in modo casuale le loro strutture sono similmente bilanciate, ma l'algoritmo di inserimento degli ARN è computazionalmente più complesso di quello degli ABR in quanto nella sua esecuzione è compreso un processo di bilanciamento e ricolorazione dei nodi. Le prestazioni dell'operazione di ricerca per alberi generati su ordine di inserimento casuale sono uguali per entrambe le strutture, in quanto alberi similmente bilanciati sui quali viene applicato lo stesso algoritmo di ricerca.