Laboratorio di Algoritmi e Strutture Dati
2020/2021 — Seconda parte

Mattia Bonaccorsi — 124610 – bonaccorsi.mattia@spes.uniud.it
Muhamed Kouate — 137359 – kouate.muhamed@spes.uniud.it
Enrico Stefanel — 137411 – stefanel.enrico@spes.uniud.it
Andriy Torchanyn — 139535 – torchanyn.andriy@spes.uniud.it

12 maggio 2021

Indice

1 Alberi binari di ricerca semplici

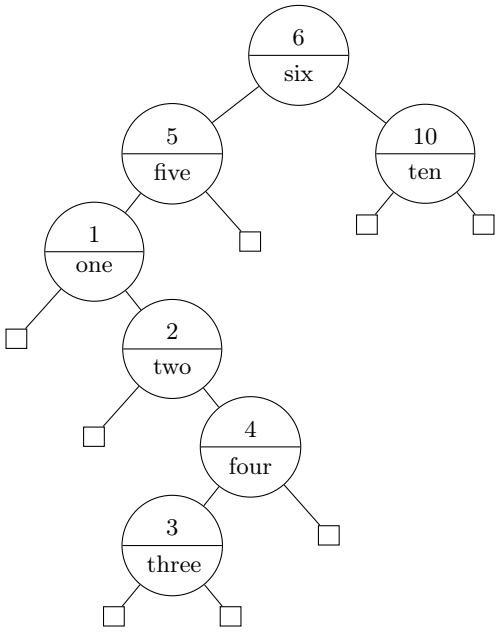
1.1 Definizione di *BST*

Un *albero binario di ricerca* (o BST) T è una struttura dati ad albero, in cui valgono le seguenti proprietà:

$$\begin{aligned} \forall x \in T,\ \forall y \in left(T) \rightarrow y.key < x.key \\ \forall x \in T,\ \forall z \in right(T) \rightarrow z.key > x.key \end{aligned} \tag{$\star$}$$

dove $k.key$ indica il valore della chiave di k, e $left(B)$ (rispettivamente $right(B)$) indica il sotto-albero sinistro (rispettivamente destro) di B.

Esempio Un *BST* di tipo semplice, in cui ogni nodo contiene una chiave numerica dell'insieme $\{1, 2, 3, 4, 5, 6, 10\}$ e un campo alfanumerico di tipo stringa, è il seguente:



Bisogna notare che non è l'unico *BST* costruibile partendo dallo stesso insieme di chiavi. Un'alternativa, per esempio, potrebbe essere stata quella di utilizzare il valore minore come chiave per la radice dell'albero, e attaccare in ordine crescente le altre chiavi, ognuna come figlio destro del nodo precedente.

2 Alberi binari di ricerca di tipo AVL

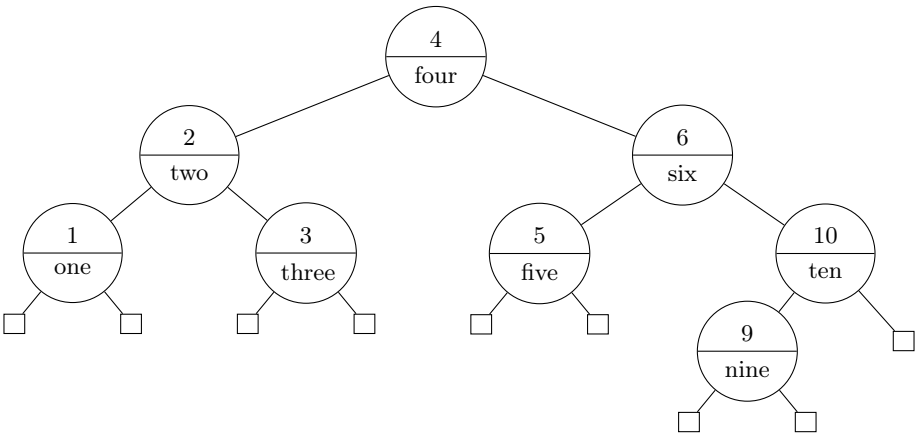
2.1 Definizione di Albero *AVL*

Un *albero AVL* T è un *BST* ($\star$), in cui vale la seguente proprietà:

$$\forall x \in T \rightarrow |h(left(x)) - h(right(x))| \leq 1 \qquad (*)$$

dove $h(k)$ indica il valore dell'altezza dell'albero radicato in k, e $left(B)$ (rispettivamente $right(B)$) indica il sotto-albero sinistro (rispettivamente destro) di B.

Esempio Un Albero *AVL* in cui ogni nodo contiene una chiave numerica dell'insieme $\{1, 2, 3, 4, 5, 6, 9, 10\}$ e un campo alfanumerico di tipo stringa, è il seguente:



, dove, ad esempio, $left(\texttt{root})$ ha altezza 2, mentre $right(\texttt{root})$ ha altezza 3.

3 Alberi binari di ricerca di tipo Red-Black

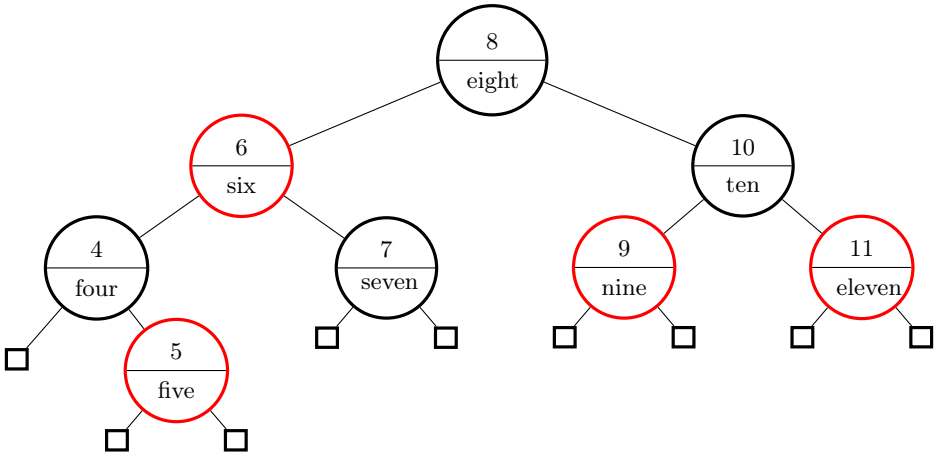
3.1 Definizione di *RB Tree*

Un *albero di tipo Red-Black* (o *RB Tree*) T è un *BST* ($\star$), in cui ogni nodo ha associato un campo "colore", che può assumere valore *rosso* o *nero*, ed inoltre vale che:

$$\forall x \in T \rightarrow h_b(left(x)) = h_b(right(x)) \qquad (\bullet)$$

dove $h_b(x)$ indica l'altezza nera dell'albero radicato in x, ovvero il massimo numero di nodi neri lungo un possibile cammino da x a una foglia.

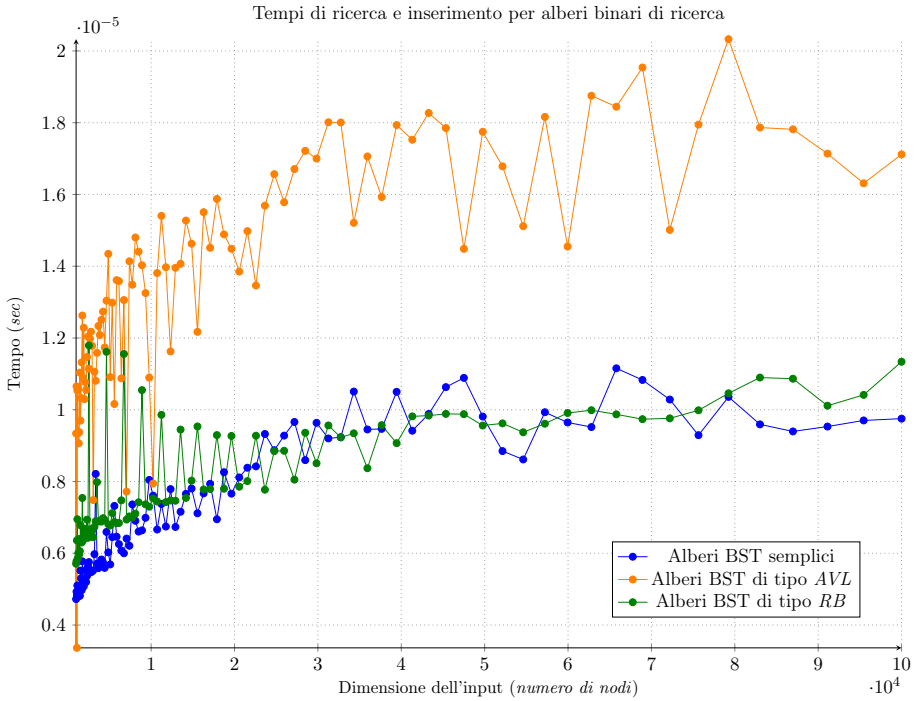
Esempio Un *BST* di tipo Red-Black, in cui ogni nodo contiene una chiave numerica dell'insieme $\{4, 5, 6, 7, 8, 9, 10, 11\}$ e un campo alfanumerico di tipo stringa, è il seguente:

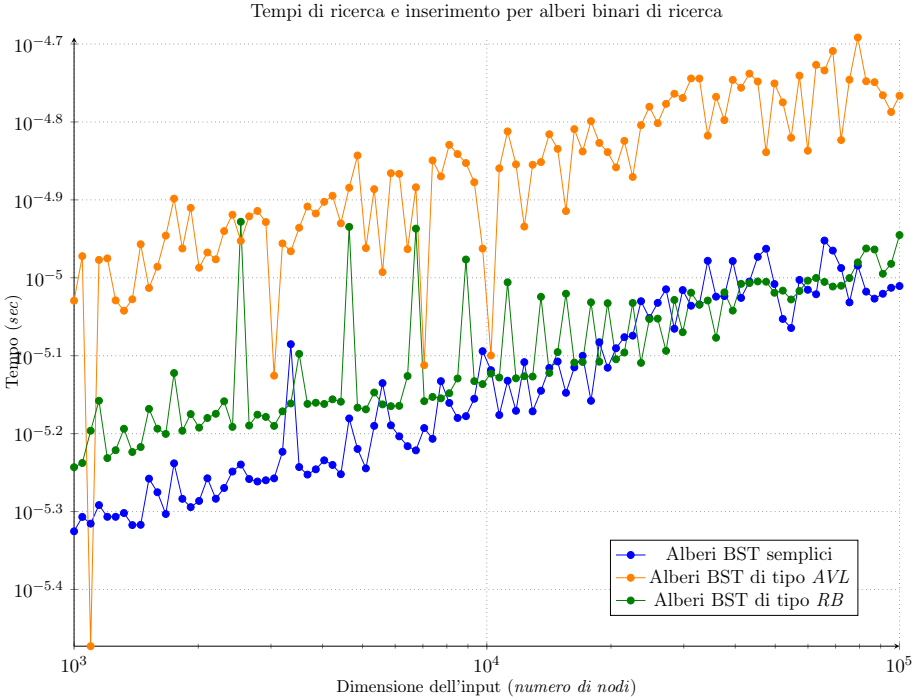


4 Calcolo della complessità

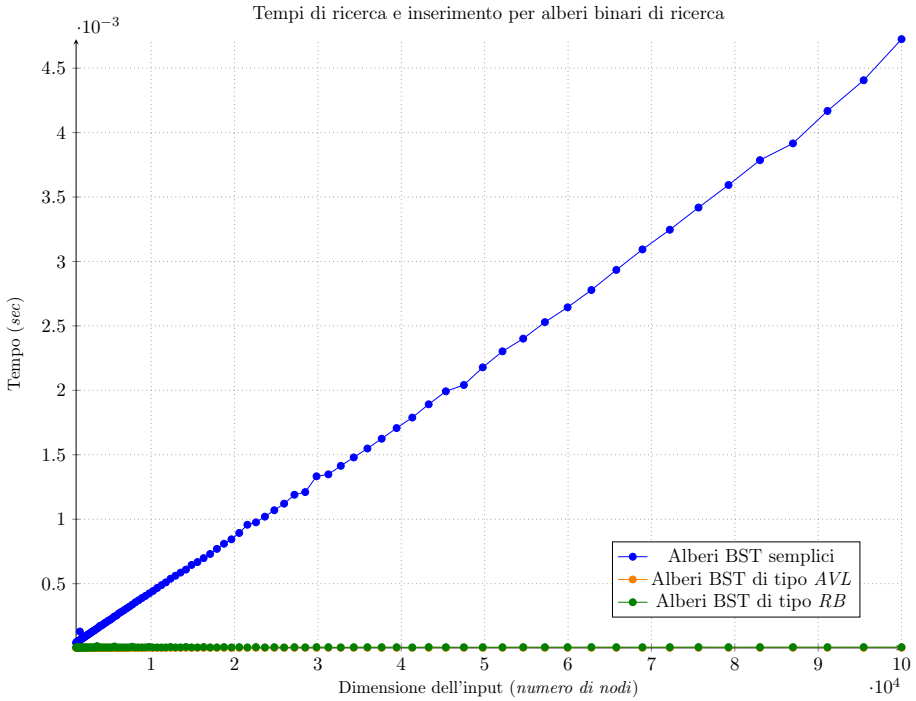
Implementate le tre strutture dati precedentemente descritte utilizzando il linguaggio Python, si è poi proceduto a calcolare i tempi medi per la ricerca e l'inserimento di n chiavi generate in modo pseudo-casuale.

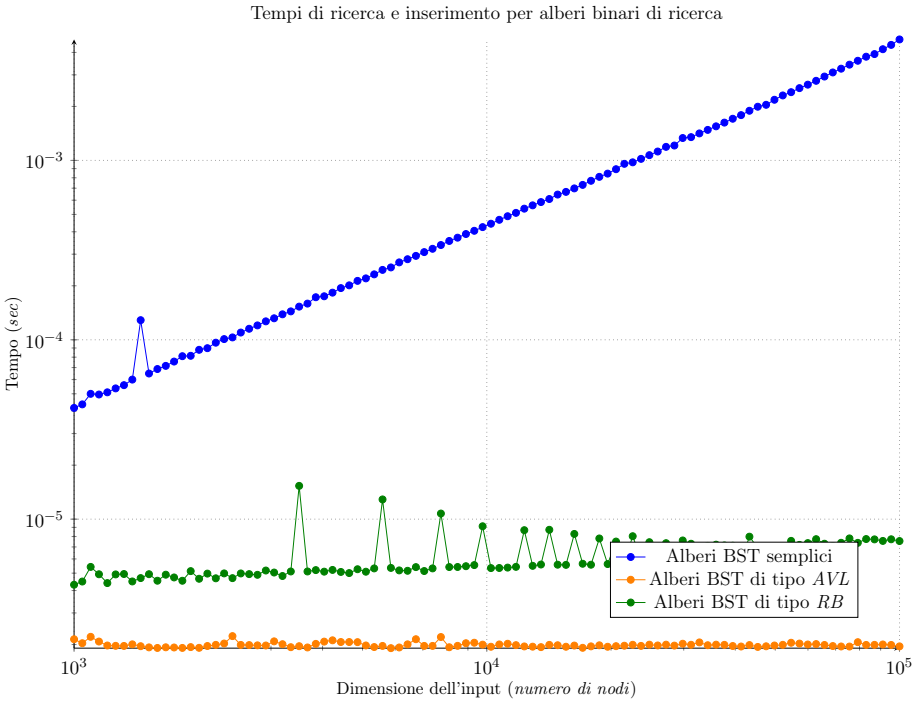
4.1 Caso random





4.2 Caso sorted





4.3 Caso smart