

Relazione struttura dati

Skip-List

La scelta del valore `max_height`, che rappresenta l'altezza massima consentita per un elemento nella skip-list, ha un impatto significativo sulle prestazioni delle operazioni di inserimento e ricerca. La probabilità di aumentare il livello di un nodo influisce sulla distribuzione degli elementi all'interno della struttura.

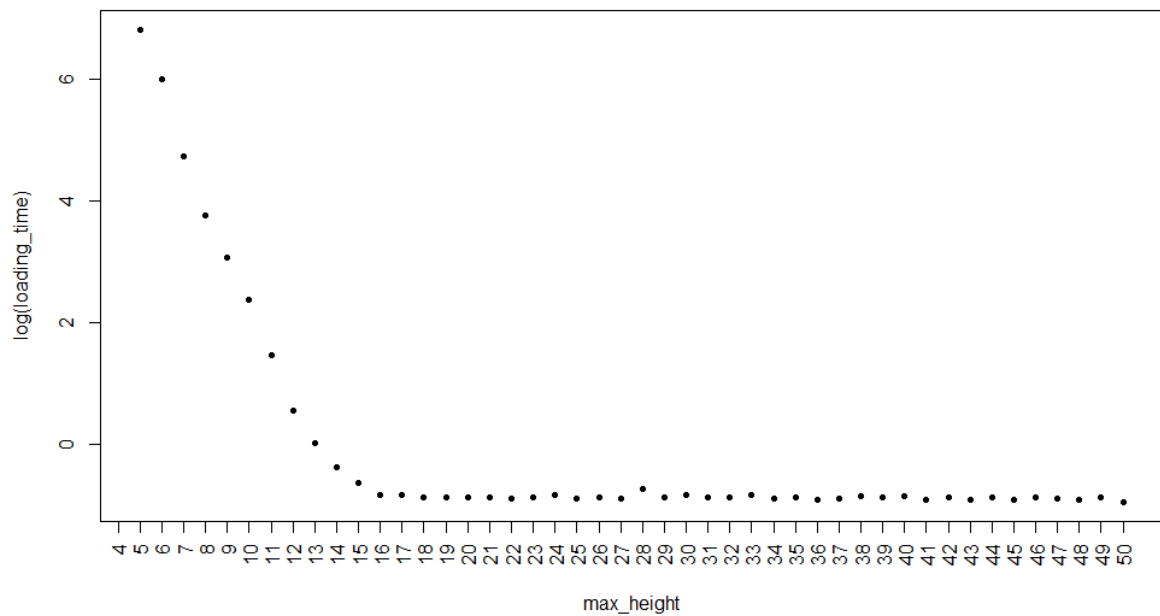
Supponendo una probabilità semplice come 1 su 2 (50%) di aumentare il livello di un nodo, possiamo esaminare tre casi possibili nella scelta del valore `max_height`.

Caso 1 (`max_height` tra 1 e $\log_2(nitems)$): In questo caso, le prestazioni della struttura dati sono degradate. Matematicamente, non sarà possibile ottenere una velocità di inserimento e ricerca $\log N$, in quanto la time complexity varia tra N (`max_height` = 1) e $\sim \log N$ (`max_height` < $\log_2(nitems)$).

Caso 2 (`max_height` = $\log_2(nitems)$): Questa è la soluzione ottimale dal punto di vista matematico. La struttura dati raggiunge prestazioni di inserimento e ricerca $\log N$.

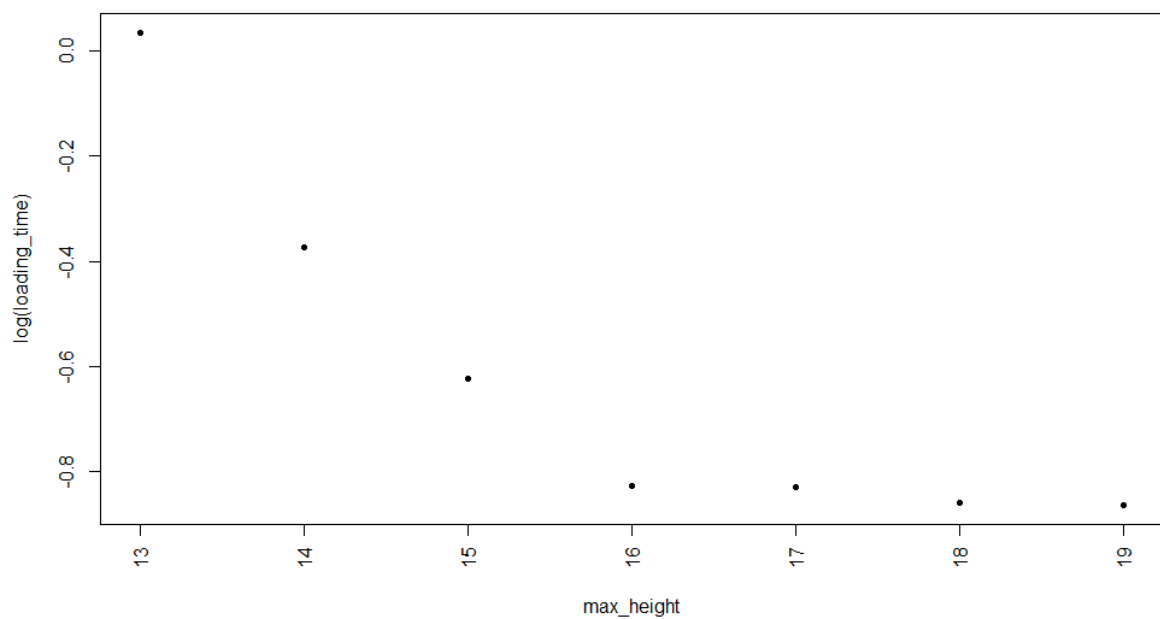
Caso 3 (`max_height` > $\log_2(nitems)$): Aumentare il valore di `max_height` oltre il valore ottimale non porta alcun vantaggio. Le prestazioni di inserimento e ricerca rimangono comunque $\log N$.

Possiamo verificare quanto detto utilizzando i dati forniti dal docente nel file `dictionary.txt`, rappresentandoli in un grafico con il tempo di caricamento dei dati (in scala logaritmica) sull'asse y e il valore di `max_height` sull'asse x . La quantità di dati processati è costante, circa 661562 elementi.



Osserviamo che il tempo di inserimento segue una tendenza inversa dell'esponenziale ($1/x^2$) verso il valore $\text{Log}_2(661562) \approx 19$, dopo il quale si stabilizza.

Possiamo vedere in dettaglio il comportamento dei valori nel grafico qui sotto.



In conclusione, possiamo affermare che il valore ottimale per max_height nell'esperimento è 19, ma già a partire da 16 si nota una differenza di prestazioni minima.