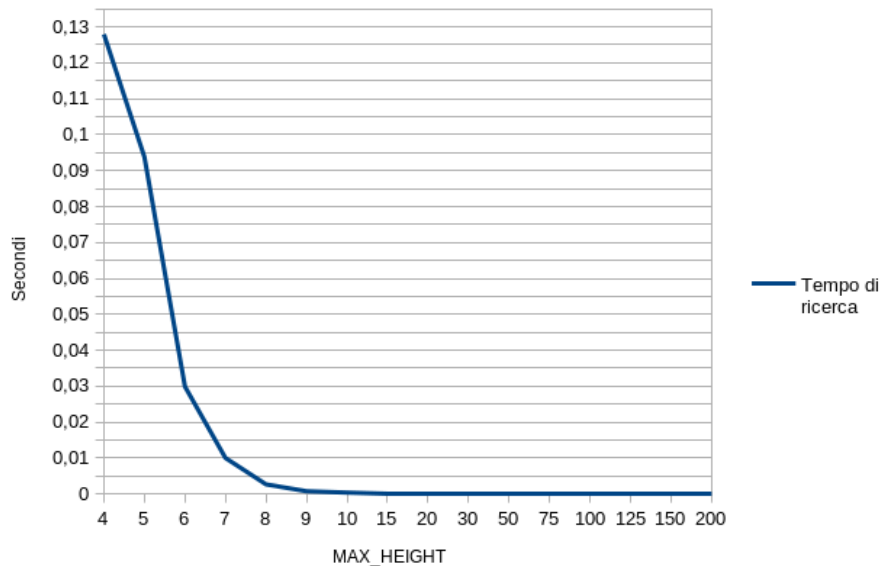
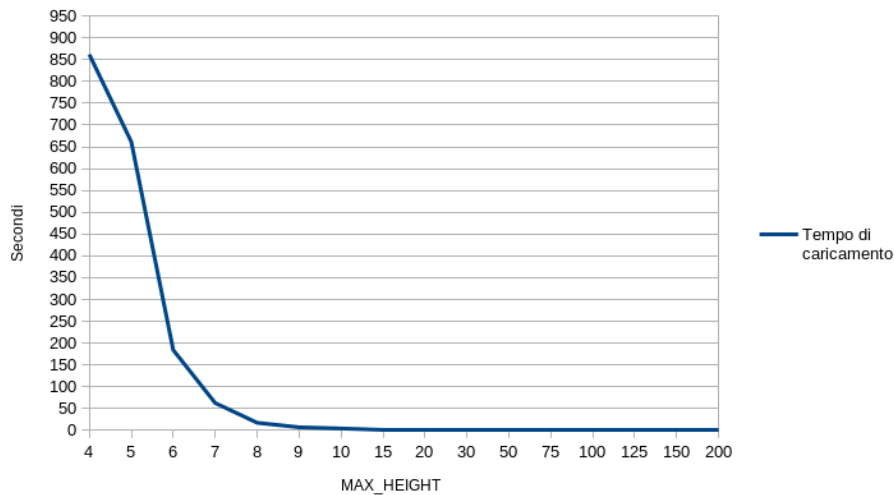


Esercizio 2

Algoritmi implementati: **ricerca su una skiplist**.

In questo esercizio si sfrutta la velocità di ricerca e l'ordinamento che può offrire una skiplist, ovvero una struttura dati a lista dove ogni nodo contiene da 1 a n puntatori che punteranno ai nodi successivi.

A seguito dei test abbiamo riscontrato una interessante differenza sia nei tempi di ordinamento nella creazione della skiplist (nel nostro caso il dizionario) che di ricerca sulle differenti altezze:



Come si può notare c'è una stretta proporzionalità in termini di tempi sia nella costruzione della skiplist che nella ricerca.

Se non ci fosse una funzione che generasse altezze casuali fino alla massima impostata, avremmo salti possibili **a non più** del valore dato in ingresso alla funzione. Nel nostro caso, quindi, la funzione "random_level()" ci permette di avere più variazione di livelli contenuti nei nodi consentendoci di visitarne meno durante la ricerca del valore desiderato portandoci alle seguenti conclusioni:

- sopra “MAX_HEIGHT “ **con valore 15** non si ha variazione dei risultati perché già solo la probabilità di avere un nodo con altezza di 15 è di $(1/2)^{15} = 0,000030518$, quindi risulta improbabile avere nodi con altezza superiore;
- riguardo, invece, a “MAX_HEIGHT “ **con valori inferiori a 10** si ha sempre di più un’inefficienza per via della poca variazione di altezze nei vari nodi;
- **con valori inferiori a 4** la creazione impiega più di 15 minuti in quanto si avvicina sempre di più all’implementazione di una classica lista con scorrimento sequenziale, quindi con complessità $O(n^2)$;
- “MAX_HEIGHT “ **di altezza 1** è come una semplice lista.