

Relazione Algoritmi e Strutture Dati

Eduard Antonovic Occhipinti, Iman Solaih, Marco Molica

April 29, 2022

Contents

0.1	Quick Sort	2
0.1.1	Impatto della scelta del pivot nel quick sort	2
0.2	Binary Insertion Sort	12
0.3	Skip List	12
0.4	Minimum Heap	12
0.5	Graph	12

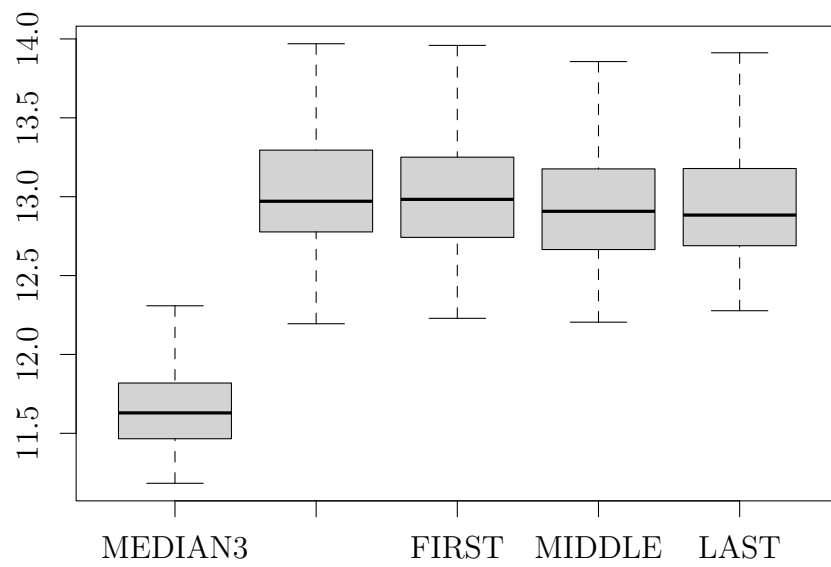
Esercizio 1

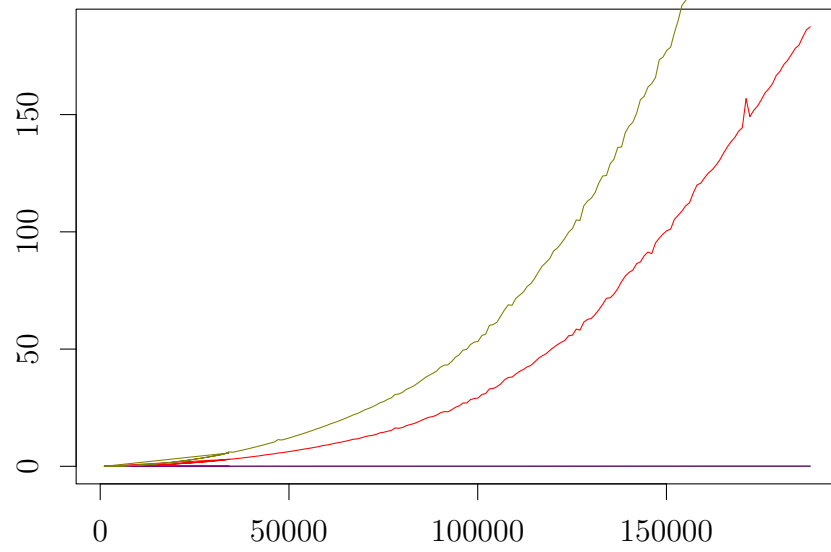
0.1 Quick Sort

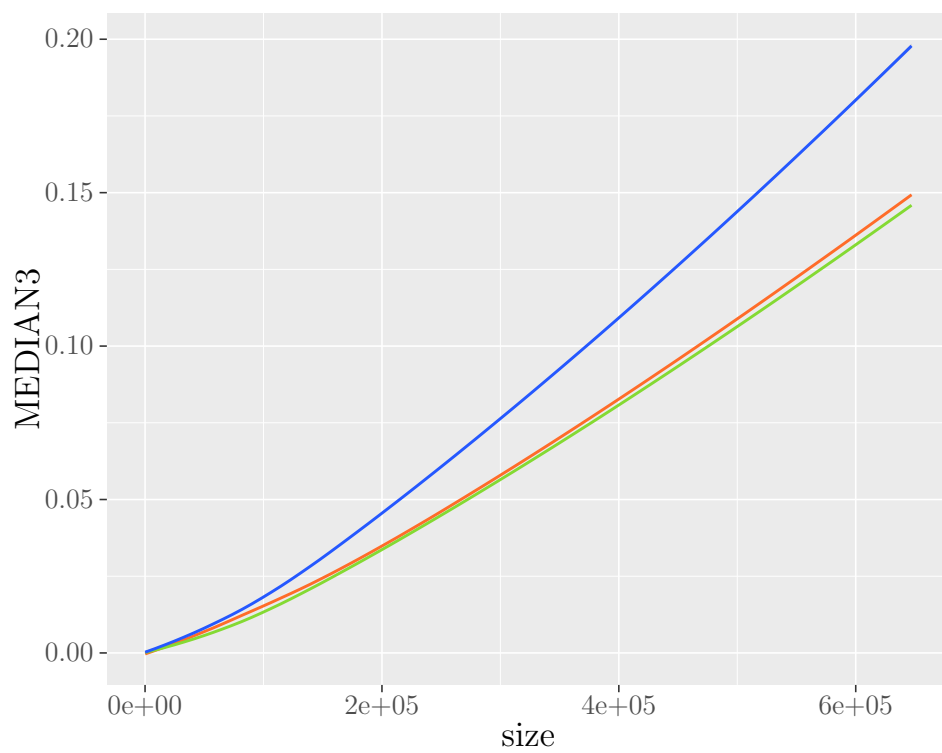
Il `quick_sort()` è un algoritmo che ordina una collezione partendo da un pivot, il pivot può essere scelto in vari modi, e in base a quale viene scelto il tempo di sorting varia. Il `quick_sort()` utilizza `_part()` per scegliere il pivot prima di chiamare `partition()` per dividere gli elementi del range selezionato in un sottoinsieme di elementi maggiori e uno di elementi minori del pivot la cui posizione finale viene restituita dal metodo.

0.1.1 Impatto della scelta del pivot nel quick sort

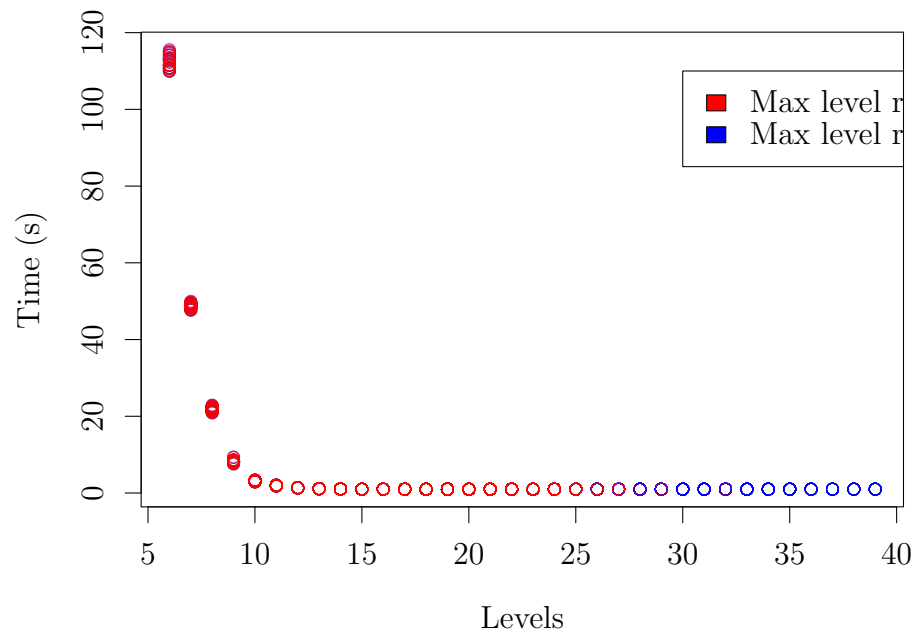
La chiamata a `rand()` porta il `quick_sort()` con pivot scelto randomicamente o come mediana di tre numeri ad essere mediamente più lento rispetto agli altri 3 casi presi in considerazione. La tabella sottostante riporta il tempo impiegato ad ordinare un array di 20 milioni elementi di tipo `struct Record`

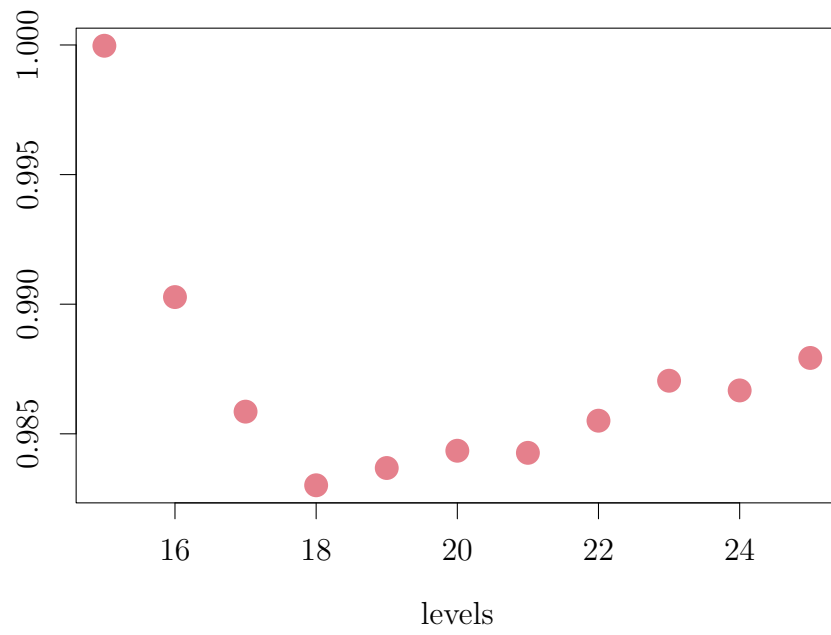




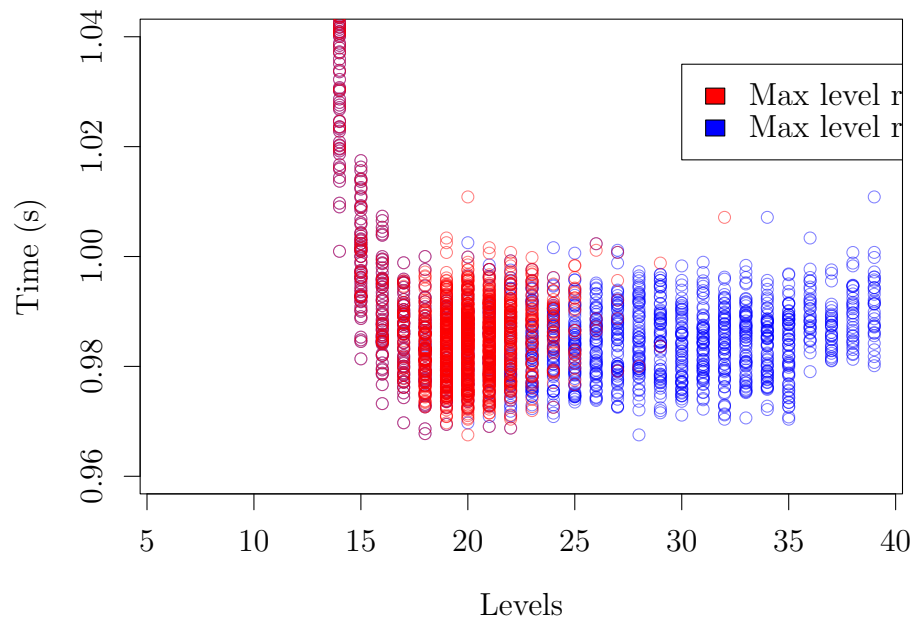


Skip List Insert

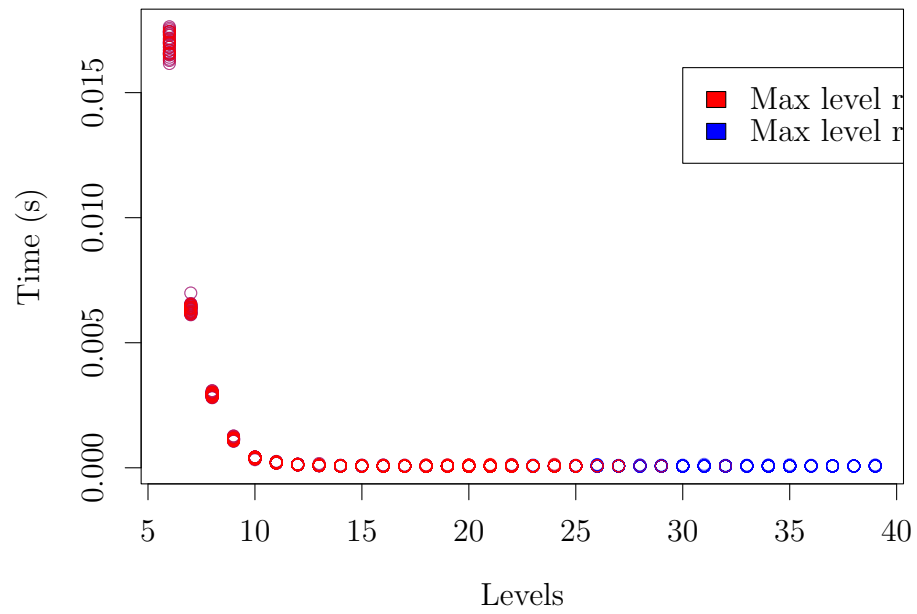


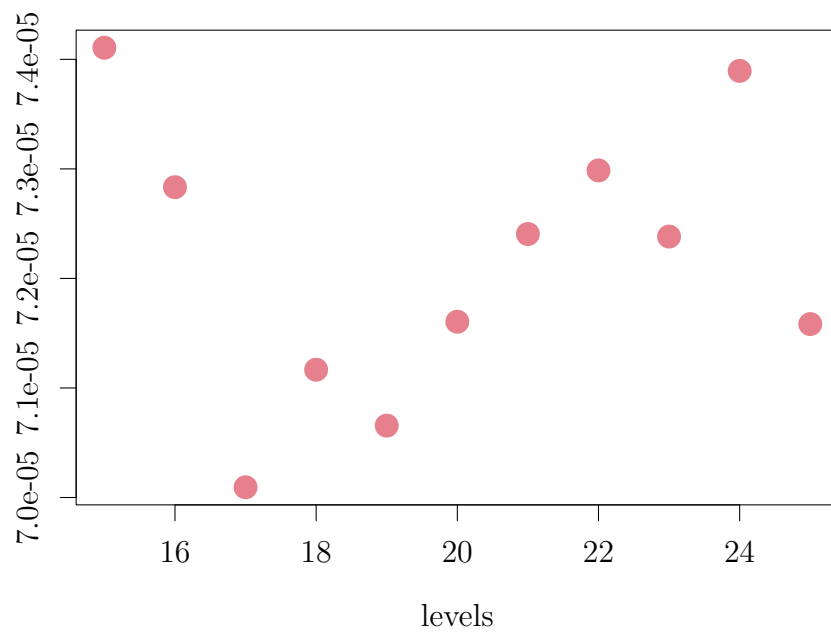


Skip List Insert

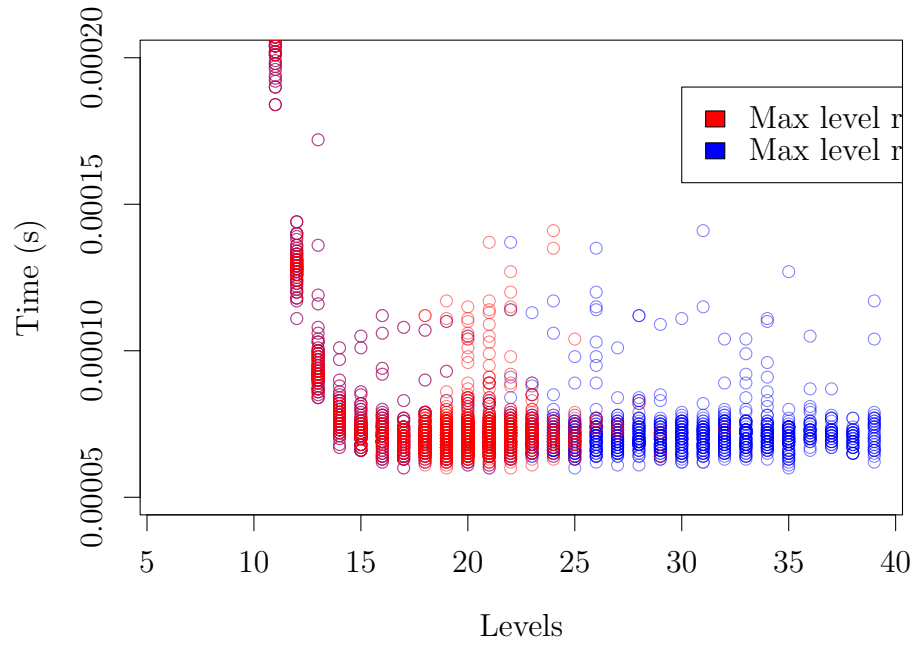


Skip List Search





Skip List Search



Esercizio 2

0.2 Binary Insertion Sort

0.3 Skip List

0.4 Minimum Heap

0.5 Graph