

# Relazione sulla scelta del pivot nel Quicksort

Matteo Barone, Stefano Cipolletta

14 Giugno 2022

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Tempi d'esecuzione</b>	<b>3</b>
2.1	Confronto tra record di lunghezza diversa . . . . .	3
2.2	Confronto tra diversi pivot . . . . .	4
<b>3</b>	<b>Conclusioni</b>	<b>5</b>

# **1 Introduzione**

La scelta del pivot è una parte fondamentale nel processo di sviluppo dell'algoritmo Quicksort.

Di seguito andremo quindi ad analizzare come la posizione del pivot e la lunghezza del record influenzano le prestazioni.

## 2 Tempi d'esecuzione

### 2.1 Confronto tra record di lunghezza diversa

Come prima cosa confrontiamo i tempi di esecuzione aumentando il numero di record e fissando come pivot il primo elemento dell'array.

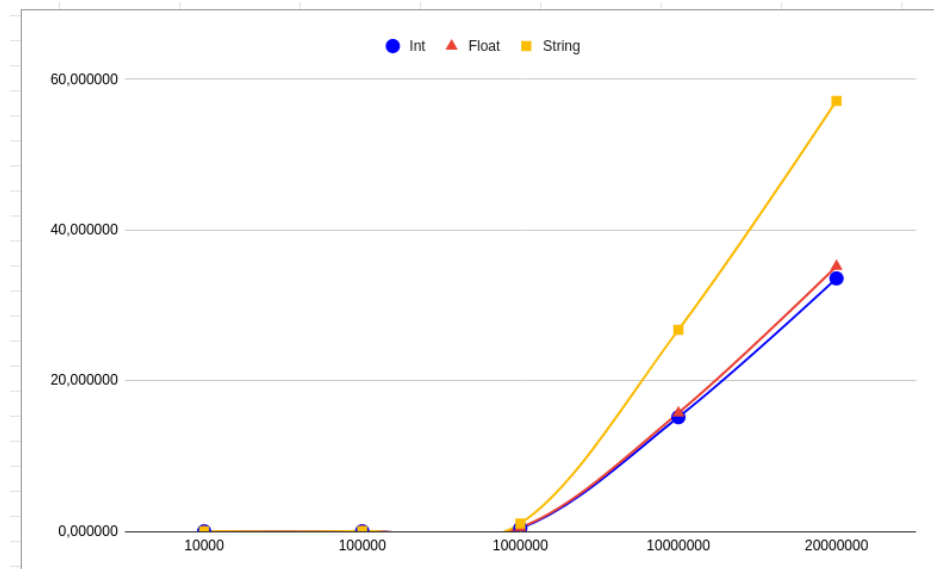
#	Int	Float	String
10000	0	0	0.015625
100000	0.015625	0.015925	0.03125
1000000	0.40625	0.51875	1.04
10000000	15.15625	15.724534	26.75
20000000	33.578125	35.173454	57.156246

Possiamo notare che con un numero di record basso, la differenza tra Int/Float e String è irrilevante.

Tuttavia, già con un record di 10 milioni si inizia a notare una grossa differenza in tempi di esecuzione.

Con 20 milioni la differenza è decisamente rilevante in tempi assoluti. Mantiene tuttavia il ratio delle precedenti prove con meno elementi. Il quicksort infatti impiega circa il doppio del tempo nel riordinare le stringhe rispetto agli Int o i Float.

Questo fenomeno è probabilmente causato dalla differenza nel operatore di confronto <sup>1</sup>.



<sup>1</sup>== per i numeri, strcmp() per le stringhe

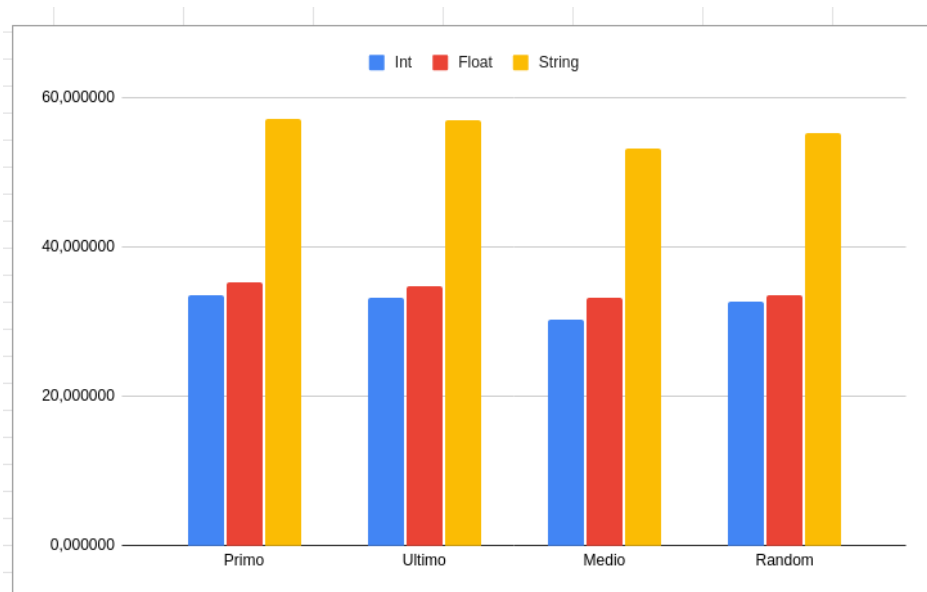
## 2.2 Confronto tra diversi pivot

La scelta del pivot è un altro fattore chiave durante lo sviluppo dell'algoritmo Quicksort.

Utilizzando 20 milioni di record, otteniamo i seguenti risultati:

#	Int	Float	String
Primo	33.578125	35.173454	57.156246
Ultimo	33.104234	34.781643	56.875245
Medio	30.254634	33.143456	53.145434
Random	32.675345	33.456743	55.143569

Come possiamo notare scegliere il primo o l'ultimo pivot è indifferente. Con il pivot centrale le prestazioni migliorano, anche se non di molto. Un pivot randomizzato ha prestazioni variabili ovviamente, ma è comunque migliore del pivot per primo/ultimo.



### **3 Conclusioni**

La scelta del pivot ha una certa influenza sulla velocità del Quicksort. Tuttavia ciò è apprezzabile solo con un array di lunghezza elevata, in questo caso da 10 milioni in su.