Arianna Lezzi 947010, Fabio Giordana 886377, Samuel Gonçalves Barbosa 955864

**Scelta del Pivot nel Quick Sort**

Per studiare il comportamento del *Quick Sort* variando la scelta del *Pivot* sono stati implementati due algoritmi differenti:

* *Quick Sort* in cui il *Pivot* è il primo elemento e il partizionamento avviene con il *three-way partition*;
* *Randomized Quick Sort* in cui la scelta del *Pivot* per ogni partizionamento è casuale (il metodo di partizionamento usato è sempre il *three-way partition*).

Si riportano i tempi calcolati ordinando il file “records.csv”:

*Quick Sort*:

* stringhe 17.94 s
* Interi 13.98 s
* float 14.84 s

*Randomized Quick Sort:*

* stringhe 17.84 s
* interi 13.55 s
* float 15.32 s

Nel caso medio non si riscontrano differenze sostanziali tra i due algoritmi.

Dato che il *Quick sort* ha come caso peggiore l’ordinamento di un array già ordinato, ci si aspettava che la scelta randomica del Pivot desse migliori risultati in questo caso, evitando di utilizzare come Pivot un elemento già nella posizione corretta e che avrebbe creato due array squilibrati nel partizionamento.

*Quick Sort* caso peggiore(campi già ordinati):

* stringhe 87.93 s
* interi >10 min
* float >10 min

*Randomized Quick Sort* caso peggiore:

* stringhe 65.44 s
* interi 95.12 s
* float 125.66 s

Nel caso degli interi e dei float si è verificato quanto ci si aspettava. Per le stringhe anche il quick sort standard termina: questo probabilmente è dovuto al metodo di partizionamento utilizzato che esclude gli elementi uguali al Pivot da chiamate successive del Quick Sort; questo è particolarmente significativo per le stringhe dato che sono presenti molti elementi uguali ripetuti.