**Progetto Algoritmi e Strutture Dati**

**Esercizio 1 – Analisi sulla scelta del pivot**

|  |  |  |  |
| --- | --- | --- | --- |
| Posizione Pivot | Interi | Stringhe | Float |
| Start array (low) | ~0 sec | ~15 sec | ~1 sec |
| End array (high) | ~0 sec | ~16 sec | ~0 sec |
| Mid of array ((low+high)/2) | ~0 sec | ~1 sec | ~0 sec |

**Esaminiamo prima di tutto nel dettaglio 2 casi.**

1. **Start array**, ossia quando scegliamo come pivot l’elemento piu’ a sinistra dell’array; in questo modo il caso peggiore è quando l’array su cui ordiniamo è già nello stesso ordine ricercato;
2. **End array**, ossia quando scegliamo come pivot l’elemento piu’ a destra dell’array; in questo modo il caso peggiore è quando l’array su cui ordiniamo è disposto nell’ordine opposto;

Complessivamente, essi aumentano la probabilità di incorrere nel caso peggiore dove la complessità è theta(n^2), cosa che non accade quando scegliamo l’elemento centrale come pivot.

Ipotizziamo, quindi, l’array ordinato e come pivot l’elemento centrale, ne risulta che la parte [pivot,.., n-1] rientra nel caso peggiore, a differenza della restante parte dell’array. Specularmente lo stesso ragionamento si può fare con l’array nell’ordine opposto.

Abbiamo testato il nostro codice con 1000000 elementi di input, le tempistiche risultanti rispettano quanto detto precedentemente.