



Bij het plotten van de data kan men kiezen voor een loglog plot (waarbij beiden assen op logaritmische schaal staan) en een (normale) lineaire plot. De loglog plot heeft het voordeel dat het de graad van de complexiteit duidelijk weergeeft in de richtingscoëfficiënt van de grafiek, in dit geval is deze plot niet echt een meerwaarde voor de analyse van de tijdscomplexiteit. De lineaire plot heeft dan weer als voordeel dat het intuïtiever is voor een mens om in te schatten hoe de tijdscomplexiteit evolueert als functie van de invoer.

We kozen ervoor om arrays te onderzoeken met een lengte tot en met 10000. Enerzijds wil men zo groot mogelijke arrays onderzoeken aangezien dit meetfouten relatief verkleint. Anderzijds wil men de uitvoeringstijd van het programma haalbaar houden. 10000 leek ons daarbij een goed compromis. Omdat quicksort ook sneller is kan het grotere lijsten aan.

Zoals voorheen, sorteerde de lijsten met integers sneller dan die met floating point getallen.

De geplotte data gedraagt zich zo goed als verwacht. Tegen onze verwachting in leunt Hoare dichter aan bij de average case terwijl Lomuto eerder naar de best case neigt. We hadden eerder verwacht dat ze dichter bij elkaar gingen liggen en zo ook dichter bij de average case.

Net als bij de andere algoritmes zijn de lagere orde termen verwaarloosbaar voor grote N .

Het zou nuttig zijn meerdere experimenten per grootte van array uit te voeren, maar aangezien we genoeg datapunten beschouwen zien we de afwijking visueel en kunnen we hier manueel abstractie van maken.

Aangezien de worst case voor quicksort $\sim \frac{n^2}{2}$ is, zouden de datapunten niet meer leesbaar zijn als deze ook nog werd geplot. Omdat we de worst case niet plotten, is de spreiding minder duidelijk maar nog steeds intuïtief door de best case $\sim n * \log_2 n$ en average case $\sim 1.39n * \log_2 n$ plot. De worst case komt echter met een kans van $\frac{2^n}{n!}$ voor, wat voor grote n zo goed als nul is en verwaarloosbaar is.