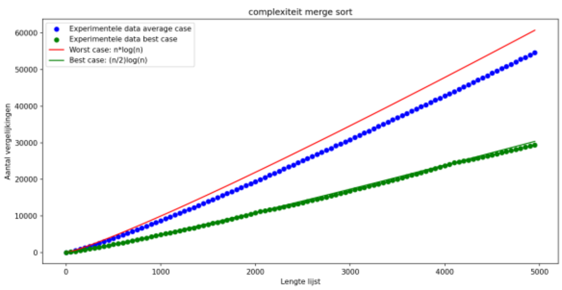

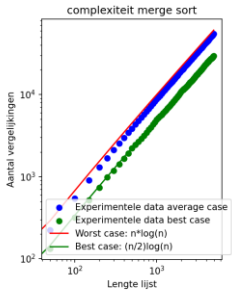

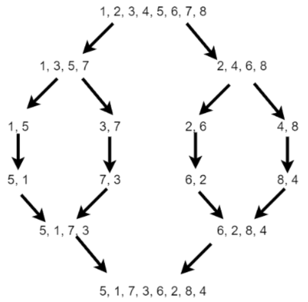

(source: https://www.baeldung.com/cs/merge-sort-time-complexity)

Bij het plotten van de data kan men kiezen voor een loglog plot (waarbij beiden assen op logaritmishe schaal staan) en een (normale) lineare plot. De loglog plot heeft het voordeel dat het de graad van de complexiteit duidelijk weergeeft in de richtingscoefficient van de grafiek. De lineare plot heeft dan weer als voordeel dat het intuitiever 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 5000. Enerzijds wil men zo groot mogelijke arrays onderzoeken aangezien dit meetfouten relatief verkleint. Anderzijds wil men de uitvoeringstijd van het programma haalbaar houden. 5000 leek ons daarbij een goed compromis.

Op Simon zijn pc duurde integers 97.0910 seconden (voor langere arrays) en floats 84.24094 seconden. We zien hier dus een substantieel verschil tussen floating point getallen en integer getallen. Echter, lieten we dezelfde experimenten lopen op Kas zijn pc, merkten we dat bij hem floats net langer duurden dan integers. Zo concluderen we dat het vergelijken van getallen hardware afhankelijk is.

In grote lijnen komen onze experimenten overeen met het theoretische model. Opvallend is wel dat heel wat best-case-datapunten zich onder de best-case-plot bevinden. Dit komt waarschijnlijk door de tilde notatie: voor grote N beschrijft de plot het best-case gedrag maar voor kleinere N geldt dit niet.

Wanneer we in bovenstaande grafiek de lagere orde termen zouden verwaarlozen, dan zouden we amper een verschil zien tussen beiden functies. Enkel in de loglog plot zou dit voor kleine lengtes van arrays een verschil geven. We kunnen besluiten dat we de lagere orde termen wel mogen verwaarlozen wanneer we grote waarden van N aannemene maar het moet zeker niet. Met grote N bedoelen we waardes waarvoor geen verschil waargenomen kan worden tussen de theorie en experimenten. Een voorbeeld van groot genoege N is 10^5.

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.

Door zowel de best-case als worst-case te plotten, kunnen we op de grafiek aflezen wat de spreiding is tussen best-case en worst-case.

Het plotten van de theoretische boven- en ondergrenzen kan helpen om de lezer inzicht te geven in hoe dicht de average case zich bevindt ten opzichte van de best- en worst case. Echter, het is zeer belangrijk in het achter hoofd te houden dat de kans dat deze scenarios voorkomen bijzonder klein is. Immers, de kans is $1/n!$ wat voor grote n zo goed als nul is. De best case is een gesorteerde rij met complexiteit $\sim\frac{1}{2}n\ log2n$. De worst case kan afgeleid worden uit bovenstaande diagram met complexiteit $\sim nlog2\ n$.