Summatieve huiswerkopdracht 4

Onderdeel 1A:

Pseudocode Merge Sort:

Hier volgt een pseudocode-voorbeeld vertrekkend van de rij {2,1,2\*,3}:

verdeel {2,1,2\*,3} in twee delen: {2,1} en {2\*,3}

verdeel {2,1} in twee delen: {2} en {1}

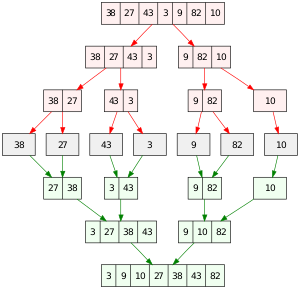
voeg {2} en {1} op volgorde samen tot {1,2}

verdeel {2\*,3} in twee delen: {2\*} en {3}

voeg {2\*} en {3} op volgorde samen tot {2\*,3}

voeg {1,2} en {2\*,3} op volgorde samen tot {1,2,2\*,3}

bron : Wikipedia, <https://nl.wikipedia.org/wiki/Mergesort#Pseudocode>



Pseudocode Merge Sort 2:

func mergesort( var a as array )

if ( n == 1 ) return a

var l1 as array = a[0] ... a[n/2]

var l2 as array = a[n/2+1] ... a[n]

l1 = mergesort( l1 )

l2 = mergesort( l2 )

return merge( l1, l2 )

end func

func merge( var a as array, var b as array )

var c as array

while ( a and b have elements )

if ( a[0] > b[0] )

add b[0] to the end of c

remove b[0] from b

else

add a[0] to the end of c

remove a[0] from a

while ( a has elements )

add a[0] to the end of c

remove a[0] from a

while ( b has elements )

add b[0] to the end of c

remove b[0] from b

return c

end func

bron : Algorithmist, <http://www.algorithmist.com/index.php/Merge_sort>

Het verschil tussen de 2 pseudocode is de mate van detail, de eerste is zeer globaal uitgelegd wat er moet gebeuren, en bij de tweede is het alleen nog maar vertalen van pseudocode naar python code. Dat zorgt ervoor dat de programmeur bij optie 2 minder denkwerk hoeft te verrichten.

Onderdeel 1B:

Voor de implementatie van Merge Sort heb ik gekozen om het te bouwen (vooral) gebaseerd op de 1ste pseudocode.

Zie mijn [Github](https://github.com/KevinGoes/Analytical-Skills) voor de implementatie!

Onderdeel 1C:

De Best case scenario is hoe minder items er in de lijst zijn natuurlijk. Apart van dat is het, het best om een even aantal items in de lijst te hebben, als er namelijk een oneven aantal items in de lijst zit moet het algoritme dat ene overgebleven item samenvoegen met een ander gesorteerde lijst. Dat betekend dat er weer een extra stap moet worden gezet. Een oneven aantal items in de lijst is dus daarbij de worst case scenario (en natuurlijk hoe meer items in de lijst).

Onderdeel 2A:

Zie wederom mijn [Github](https://github.com/KevinGoes/Analytical-Skills) voor het Lineair zoek algoritme

Onderdeel 2B:

Zie wederom mijn [Github](https://github.com/KevinGoes/Analytical-Skills) voor het Binair zoek algoritme

Als de GitHub link niet werkt: https://github.com/KevinGoes/Analytical-Skills