ALGORITHMS & DATA STRUCTURES

LES 1 : ALGORITME ANALYSE

UITWERKINGEN

THEORIE OPGAVEN

OPGAVE 1 (5.6)

- x because $O(N)$
- x^2, $x^2 - x$ because both are $O(N^2)$
- $x^3 + x$, $x^4/(x-1)$ because both are $O(N^3)$

  remark: $x^4/(x-1)$ becomes equal to x^3 for large values of x

OPGAVE 2 (5.12ACD)

a) 2 ms
b) 10 ms
c) 50 ms

OPGAVE 3 (5.16 ACD)

a) For a linear-time algorithm, we can solve a problem 120,000 times as large, or 12,000,000 (assuming sufficient resources);
b) For a quadratic algorithm, we can solve a problem $\sqrt{120000}$ = 346 times as large, so we can solve a problem of size 34,600;
c) For a cubic algorithm, we can solve a problem $\sqrt[3]{120000}$ = 49 times as large, so we can solve an instance of size 4,900.

OPGAVE 4 (5.19)

$O(N)$

OPGAVE 5 (5.26)

1) $O(N)$
2) $O(N)$
3) $O(N^2)$

4) O(N)
5) O(N^3)
6) O(N^2)
7) O(N^5)
8) O(logN)

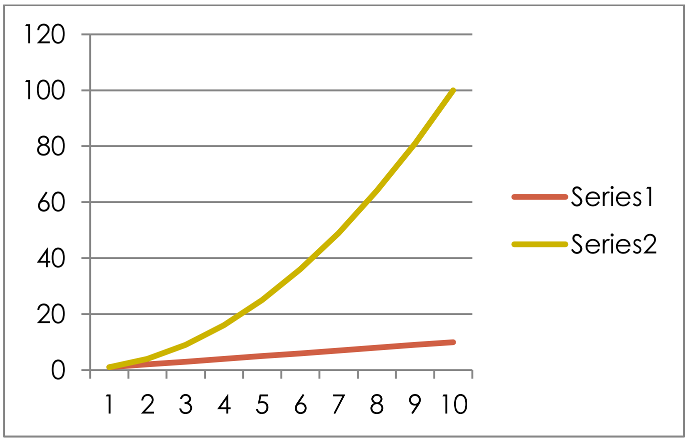
OPGAVE 6

De method `functie()` wordt in totaal N^3 aangeroepen. Binnen `functie()` wordt `functie2()` x^2 keer aangeroepen. Dus `functie2()` wordt N^5 keer aangeroepen. De compexiteit van `functie2()` is logaritmisch dus de totale complexiteit is $O(N^5 \log(N))$.

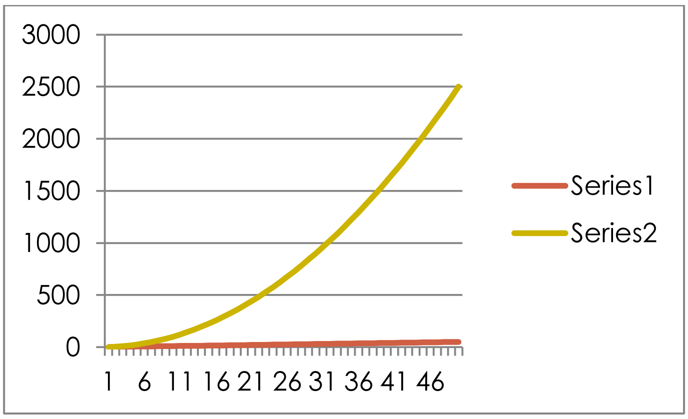
OPDRACHT 7

a)

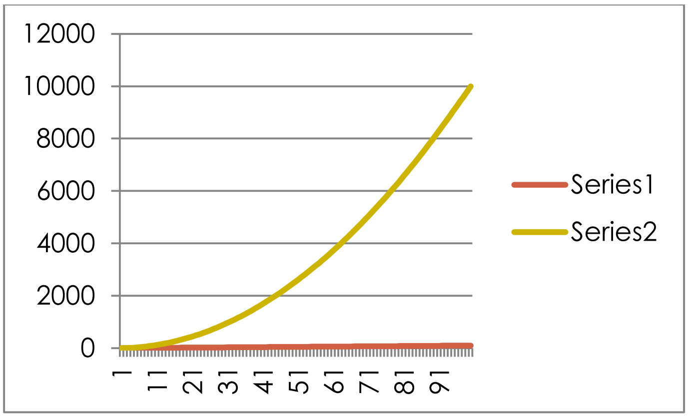
$N = 1, 2, \dots, 10$



$N \ = 1, 2, \dots, 50$



$N = 1, 2, \dots 100$



Wat opvalt is, dat voor grotere N (N.B. N is hier nog maar maximaal 100) de grafiek van N^2 al bijna wegvalt tegen de grafiek van N.

OPGAVE 8

a)

x	$\log x$
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9

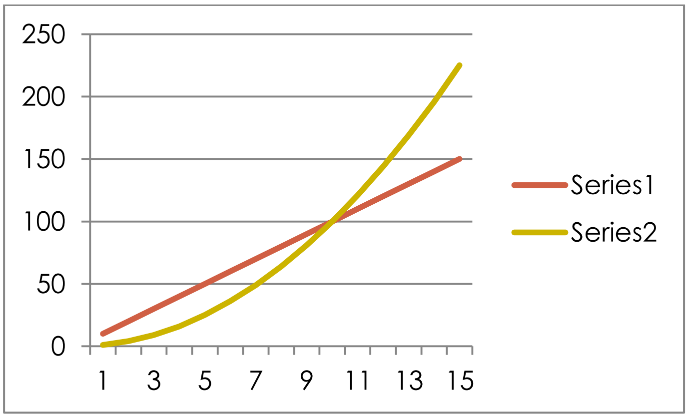
b) $\log 1024 = 10$, $\log 2048 = 11$

c) $x = 16$

d) $x = 1024$

OPGAVE 9

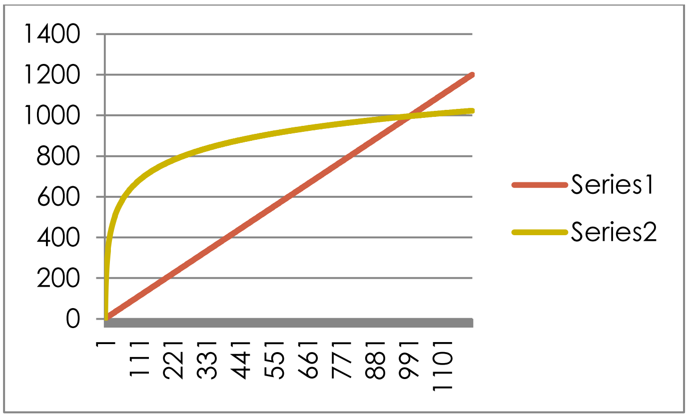
a) De grafiek laat het volgende zien:



We lezen af: de (rood getekende) kwadratische functie wordt groter dan de lineaire als x de waarde 10 heeft. De berekening toont dit aan. We zoeken de x waarvoor geldt: $x^2 = 10x$. Links en rechts delen door x levert direct op: $x = 10$. Evenzo vinden we voor $x^2 = 1000x$ als grens $x = 1000$.

Algemeen geldt het volgende. Als $x^2 = \alpha \cdot x$ met $x \neq 0$, dan volgt daaruit $x = \alpha$. Voor alle $x > \alpha$ wint de grafiek van x^2 het van die van x. Anders gezegd: voor alle $x > \alpha$ geldt dat $x^2 > \alpha \cdot x$. Daarom wint voor grote x de kwadratische functie het altijd van de lineaire, met welk getal je die lineaire ook zou vermenigvuldigen. Daarom concluderen we terecht dat voor een algoritme met complexiteit $3N + 10$ in Big-Oh notatie O(N) mag worden geschreven.

b) Uit onderstaande grafiek lezen we af: de lijn van de logaritmische functie snijdt de lijn van de lineaire functie voor $x \approx 996$.



Uit het verloop zien we, dat voor grote N de logaritmische gunstiger is dan de lineaire. Dit onderstreept de conclusie dat binair zoeken voor grote N gunstiger is dan lineair zoeken.