


11

Complexiteitsanalyse

Oefening 11.1

 Wat is de Big- O tijdscomplexiteit van onderstaand algoritme? Ga ervan out dat het print statement een constante complexiteit van $O(1)$ heeft.

```
if (n == 1) {  
    return;  
}  
  
for (int i = 1; i <= n; i++) {  
    for (int j = 1; j <= n; j++) {  
        System.out.println("A");  
        break;  
    }  
}
```

Oefening 11.2

 Wat is de Big- O tijdscomplexiteit van onderstaand algoritme?

```
int a = 0, i = n;  
while (i > 0) {  
    a += i;  
    i /= 2;  
}
```

Oefening 11.3



Wat is de Big-O tijdscomplexiteit van onderstaand algoritme?

```
int a = 0;
for (int i = 0; i < n; i++) {
    for (int j = n; j > i; j--) {
        a = a + i + j;
    }
}
```

Oefening 11.4



Wat is de Big-O tijdscomplexiteit van onderstaand algoritme? Bereken voor $k = 1$ & voor k heel groot.

```
for (int i = 0; i < n; i++) {
    i = i * k;
}
```

Oefening 11.5



Wat is de Big-O tijdscomplexiteit van onderstaand algoritme? Ga ervan uit dat het print statement een constante complexiteit van $O(1)$ heeft.

```
for (int i = 0; i < n; i++) {
    for (int j = i; j < i * i; j++) {
        if (j % i == 0) {
            for (int k = 0; k < j; k++)
                System.out.println("A");
        }
    }
}
```

Oefening 11.6



Wat is de Big-O tijdscomplexiteit van het Algoritme van Floyd zoals geïmplementeerd in Les 8?

Oefening 11.7



In een binaire boom, wat is de Big-O tijdscomplexiteit om:

1. een element te zoeken?
2. een element toe te voegen?
3. een element toe te verwijderen?

Oefening 11.8



In een binaire zoekboom, wat is de Big-O tijdscomplexiteit om:

1. een element te zoeken?
2. een element toe te voegen?
3. een element toe te verwijderen?

Oefening 11.9



In een binaire zoekboom, wat is de Big- Θ tijdscomplexiteit om:

1. een element te zoeken?
2. een element toe te voegen?
3. een element toe te verwijderen?

Oplossingen

Oplossing 11.1 $O(n)$

Oplossing 11.2 $O(\log(n))$

Oplossing 11.3 $O(n^2)$

Oplossing 11.4 voor $k = 1$: $O(n)$ en voor K heel groot: $O(\log_k(n))$

Oplossing 11.5 $O(n^5)$

Oplossing 11.6 $O(n^3)$

Oplossing 11.7

1. Zoeken: $O(n)$
2. Toevoegen: $O(n)$
3. Verwijderen: $O(n)$

Oplossing 11.8

1. Zoeken: $O(n)$
2. Toevoegen: $O(n)$
3. Verwijderen: $O(n)$

Oplossing 11.9

1. Zoeken: $O(h)$, met h de hoogte van de boom
2. Toevoegen: $O(h)$, met h de hoogte van de boom
3. Verwijderen: $O(h)$, met h de hoogte van de boom