

Flowgorithm

Standaard-algoritmes

Inleiding

Alle applicaties die door een digitaal apparaat gebruikt wordt zijn algoritmes. Er zijn een paar standaard algoritmen die belangrijk zijn om te kennen. **Zoek-algoritmes** en **sorteer-algoritmes**. Van beide zijn er verschillende ontwikkelt, elk met hun eigen voor- en nadelen en beperkingen.

Zoek-Algoritmes

Twee bekende zoekalgoritmes zijn:

1. Lineair zoeken
2. Binair zoeken.

Lineair zoeken

Dit is niet de meest efficiënte manier van zoeken. Stel je hebt een ongesorteerde verzameling van tien getallen.

Index	0	1	2	3	4	5	6	7	8	9
Waarde	8	10	3	2	7	4	9	5	6	1

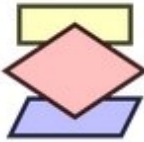
Stel dat we zoeken op welke plek in de verzameling het cijfer 7 zich bevindt. Je moet dan 5 elementen vergelijken om het cijfer 7 te vinden. Als het cijfer dat je zoekt niet voorkomt in de verzameling dan heb je alle element moeten vergelijken in de verzameling. Nu gaat het om tien elementen, maar stel dat je een naam van iemand zoekt in een telefoonboek?

De zoektijd bij lineair zoeken is dus niet constant. Het kan toevallig meteen het eerste element zijn of misschien het laatste of zelfs helemaal niet aanwezig zijn wat pas wordt gezien nadat alle element vergeleken zijn.

Het enige voordeel van lineair zoeken is dat er in een ongesorteerde verzameling gezocht kan worden.

Opdracht 1

Maak een stroomdiagram dat een lijst van 10 willekeurige gehele getallen genereerd tussen 0 en 100. Daarna deze lijst toont. Laat daarna de gebruiker een getal invoeren waarna het programma moet gaan zoeken naar dat element. Het programma moet de index aangeven waar het programma het element gevonden heeft of aangeven dat het element niet voorkomt in de verzameling.



Flowgorithm

Standaard-algoritmes

Binair zoeken

Binair zoeken is een slimmere van zoeken die over het algemeen sneller is dan lineair zoeken. Elementen worden met een na genoeg constante tijd gevonden in een verzameling.

De beperking voor binair zoeken is wel dat de verzameling gesorteerd moet zijn om te kunnen gebruiken.

Neem bijvoorbeeld de onderstaande verzameling:

Index	0	1	2	3	4	5	6	7	8	9
Waarde	10	16	23	48	51	64	75	86	89	97

1. Ga ervan uit dat de verzameling oplopend is gesorteerd (A $\rightarrow$ Z).
2. Stel we zoeken het getal 23.
3. Deel de verzameling in twee (gelijke) helften.
 - A. Vergelijk het element dat je zoekt met het laatste element in de linkerhelft.

$$23 < 51$$

Index	0	1	2	3	4	5	6	7	8	9
Waarde	10	16	23	48	51	64	75	86	89	97

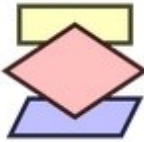
- B. Indien de twee gelijk zijn is het zoekproces klaar.
- C. Indien het element dat gezocht wordt kleiner is dan het laatste element van de linkerhelft van de verzameling dan wordt er in die helft verder gezocht. Zo niet wordt er in de rechterhelft verder gezocht.
- D. Deel de nieuwe helft opnieuw in twee gelijke delen en vergelijk het te zoeken element opnieuw met het nieuwe laatste element van de linkerhelft.

$$23 = 23$$

Index	0	1	2	3	4
Waarde	10	16	23	48	51

Gevonden op index = 2

- E. Herhaal dit proces tot het element gevonden is of dat er geen elementen meer in de verzameling zijn en dus het te zoeken element niet voorkomt in de verzameling.



Flowgorithm

Standaard-algoritmes

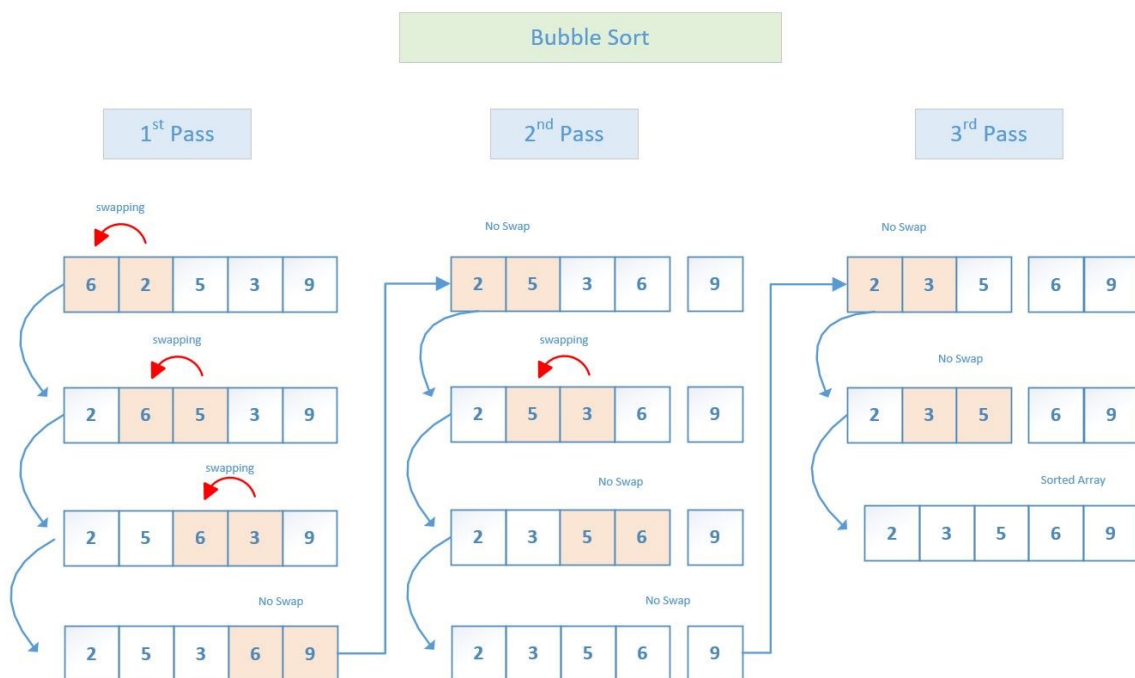
Opdracht 2

Maak een stroomdiagram dat een lijst van 10 oplopend gesorteerde gehele getallen bevat tussen 0 en 100. Daarna deze lijst toont. Laat daarna de gebruiker een getal invoeren en het programma gebruikt dan het binaire zoekalgoritme op het getal te vinden en de index aan te geven waar dat getal zich bevindt.

Sorteren

Net als voor het zoeken zijn er verschillende algoritmes bedacht voor het sorteren. Ook hier geldt dat elk algoritme zijn voor en nadelen heeft. De ene is sneller dan de andere afhankelijk van de volgorde die elementen hebben in de verzameling.

Een bekend en eenvoudig sorteer algoritme is **bubble sort**. In de afbeelding hieronder wordt getoond hoe deze manier van sorteren werkt.

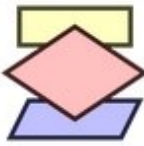


Opdracht 3

Maak een stroomdiagram in het programma Flowgorithm die een lijst van 4 elementen sorteert met het **bubble sort** algoritme.

Opdracht 4

Ga op het internet zoeken naar het **selection sort** algoritme. Beschrijf de voor- en nadelen. Leg in stappen uit hoe het algoritme werkt en maak een stroomdiagram in het programma Flowgorithm van het algoritme.



Flowgorithm

Standaard-algoritmes

Lijst zonder dubbelen

Een andere functie die vaak gebruikt wordt bij programma's is het bepalen of er dubbele elementen in een lijst zijn of het kunnen genereren van een lijst met allemaal unieke elementen.

Opdracht 5

Maak een programma die een lijst van 50 willekeurige getallen genereerd tussen 0 en 100 waarbij alle getallen maar één keer voor komt in de lijst.