

Materiaal voor de les

- Papier/Pen + Inloggen en een leeg programma openen op Khan Academy
- Tijdens oefeningen verwacht ik dat je nadenkt en probeert de flowcharts te tekenen en oefeningen te programmeren

T4.4 Sorteren

Mnr. Segers

Hoe sorteer jij?

Denk na hoe jij het sorteren van volgende lijst zou aanpakken [5, 2, 8, 1, 4]?

Veel verschillende manieren die werken, maar voor een computer moet je het stap per stap uitleggen.



Selection Sort

Wat moet je kunnen aan het einde van de les:

- Begrijpen hoe selection sort werkt
- Flowchart van selection sort begrijpen of kunnen tekenen
- De code van selection sort begrijpen



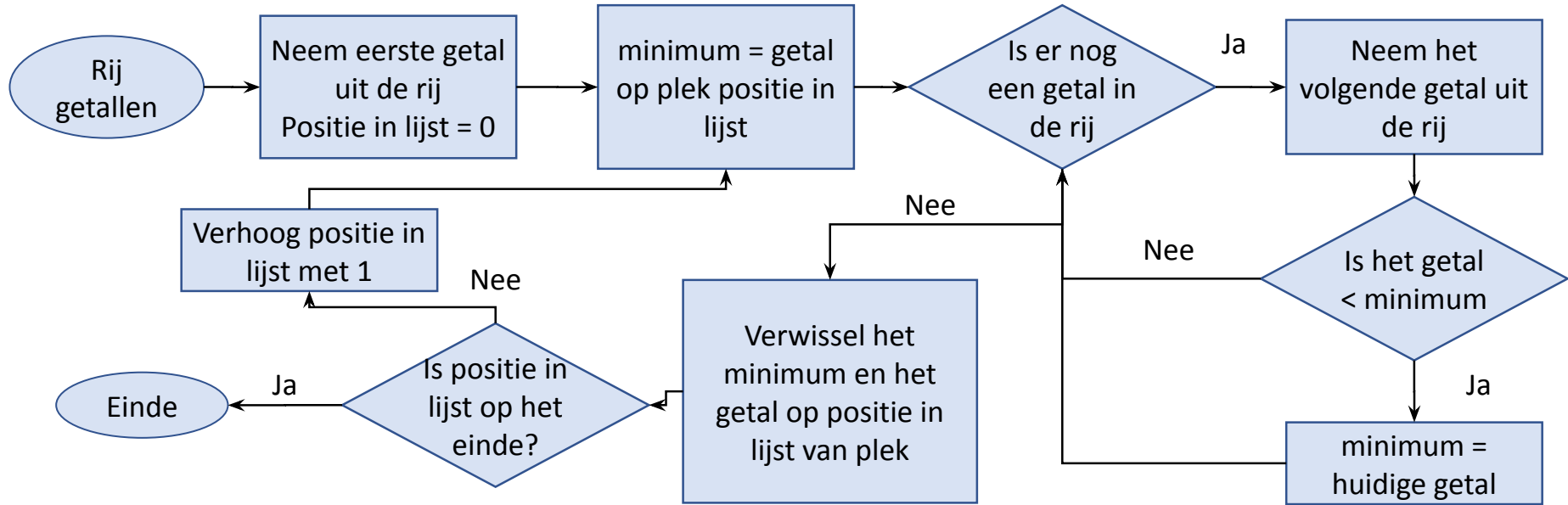
Het idee achter selection sort

- 1) Start op positie 0 in de lijst
- 2) Ga over heel de lijst om het kleinste getal te vinden
- 3) Wissel het getal op plek 0 en het kleinste getal van plaats

- 4) Start op positie 1 in de lijst
- 5) Ga over heel de lijst en zoek het kleinste getal
- 6) Wissel het getal op plek 1 en het kleinste getal van plaats

Ga zo door tot je op het einde van de lijst bent gekomen

Flowchart Selection sort



De code

- Twee for-loops in elkaar
 - Eerste om over heel de lijst te gaan
 - Tweede om kleinste in rest van lijst te vinden
- Op einde van iedere stap in eerste loop wisselen we
- Tijdelijke variabele nodig om niets kwijt te raken

```
function selectionSort(lijst) {  
    var n = lijst.length;  
  
    // Buitenste lus: doorloop elke positie van voor naar achter  
    for (var i = 0; i < n; i++) {  
  
        // Ga ervan uit dat het kleinste getal op positie i staat  
        var minIndex = i;  
  
        // Binnenste lus: zoek het echte kleinste getal in de rest  
        for (var j = i + 1; j < n; j++) {  
            if (lijst[j] < lijst[minIndex]) {  
                // Nieuw minimum gevonden  
                minIndex = j;  
            }  
        }  
  
        // Wissel het gevonden minimum naar positie i  
        var temp = lijst[i];  
        lijst[i] = lijst[minIndex];  
        lijst[minIndex] = temp;  
    }  
  
    return lijst;  
}
```

Oefening

Het sorteren van lijst = [4, 1, 3, 2]

- Elke stap is een actie:
 - Het verhogen van i
 - Het verhogen van j
 - Het aanpassen van min_index
 - Het wisselen van twee elementen in de lijst

- Probeer op het blaadje in te vullen hoe de stappen eruit zien voor deze lijst.
Er is een voorbeeld voor een andere lijst toegevoegd

VWO: Bubble Sort

Wat moet je kunnen aan het einde van de les:

- Begrijpen hoe bubble sort werkt
- Flowchart van bubble sort begrijpen of kunnen tekenen
- De code van bubble sort begrijpen



VWO: Het idee achter bubble sort

- 1) Vergelijk elementen op plek 0 en 1
- 2) Als ze verkeerd staan wissel ze om

- 3) Vergelijk elementen op plek 1 en 2
- 4) Als ze verkeerd staan wissel ze om

Na elke herhaling staat in ieder geval het grootste element op het einde
Herhaal dit net zo vaak als er elementen in de lijst staan en zo staat alles juist

VWO: Voorbeeld Bubble Sort

Voorbeeld stap voor stap op `[4, 2, 3, 1]`:

Ronde 1 — het grootste getal (4) bubbelt naar achteren:

`[4, 2, 3, 1]` → $4 > 2$, wissel → `[2, 4, 3, 1]`

`[2, 4, 3, 1]` → $4 > 3$, wissel → `[2, 3, 4, 1]`

`[2, 3, 4, 1]` → $4 > 1$, wissel → `[2, 3, 1, 4]` ✓ 4 op zijn plek

Ronde 2 — 3 bubbelt naar zijn plek:

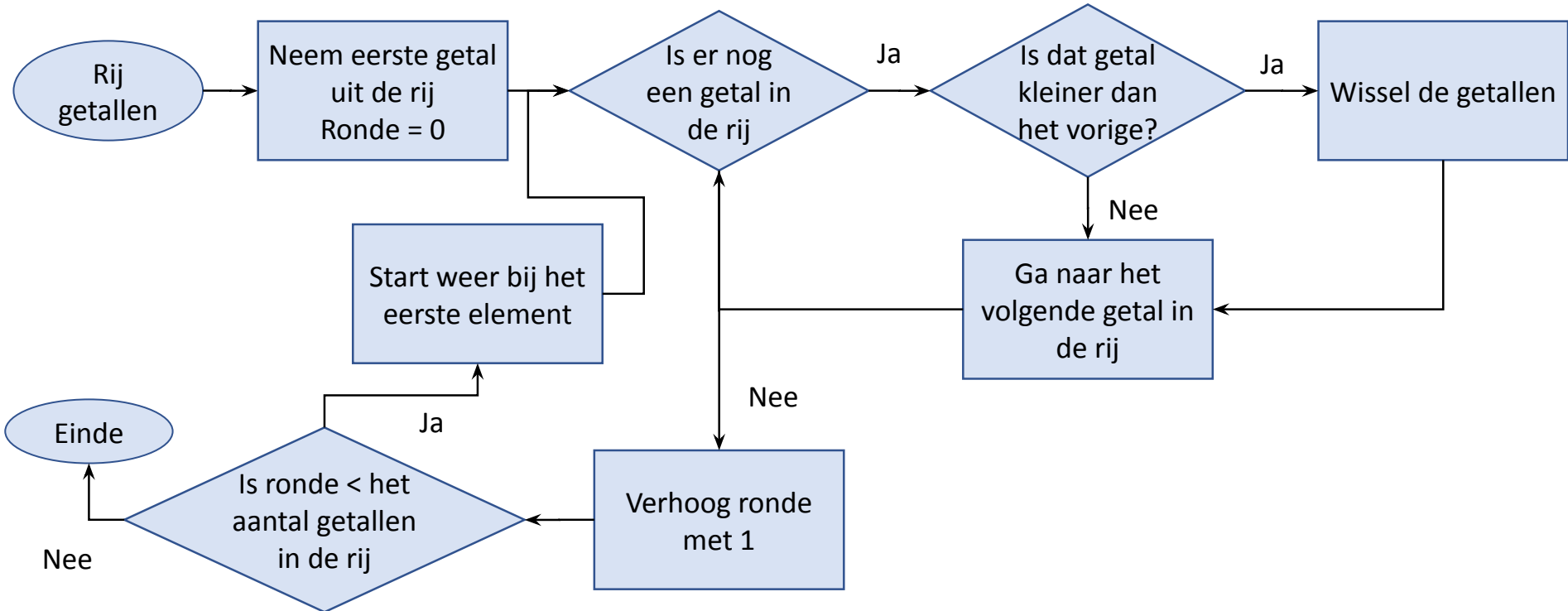
`[2, 3, 1, 4]` → $2 < 3$, geen wissel

`[2, 3, 1, 4]` → $3 > 1$, wissel → `[2, 1, 3, 4]` ✓ 3 op zijn plek

Ronde 3:

`[2, 1, 3, 4]` → $2 > 1$, wissel → `[1, 2, 3, 4]` ✓ klaar

VWO: Flowchart Bubble sort



VWO: De code

- Twee for-loops in elkaar
 - Eerste om rondes te tellen
 - Tweede om over lijst te gaan
- Binnen de tweede for-loop wisselen we heel de tijd
- Tijdelijke variabele nodig om niets kwijt te raken

```
function bubbleSort(lijst) {  
    var n = lijst.length;  
  
    // Buitenste lus: elke ronde plaatst het grootste getal achteraan  
    for (var i = 0; i < n - 1; i++) {  
  
        // Binnenste lus: vergelijk burens, stop eerder elke ronde  
        // Want na ronde i zijn de laatste i getallen al op hun plek  
        for (var j = 0; j < n - 1 - i; j++) {  
  
            // Als linker buur groter is dan rechter buur: wissel  
            if (lijst[j] > lijst[j + 1]) {  
                var temp = lijst[j];  
                lijst[j] = lijst[j + 1];  
                lijst[j + 1] = temp;  
            }  
        }  
    }  
  
    return lijst;  
}
```

VWO: Wat is je favoriet?

- Snelheid: Ongeveer even snel
 - Tenzij je check in bubble sort of er een ronde 0 wissels zijn gebeurd, dan kan je vroeger stoppen en is bubble sort sneller
- Wissels: Bubble sort wisselt veel meer
 - Kan erg traag worden als je grote data continue moet wisselen van plek

