

Eerste huiswerkopgave Complexiteit 2025

Inleveren: maandag 10 maart 2025

Maak de uitwerking in **LaTeX!**

Inleveren via Brightspace

Geef een **duidelijke toelichting/uitleg** bij al je antwoorden!

Gegeven is een array $A = A[1], A[2], \dots, A[n]$ dat $n > 2$ gehele getallen bevat. We bekijken een algoritme dat de index van de kleinste waarde oplevert. Indien de kleinste waarde meerdere keren voorkomt wordt de index van het eerste voorkomen vanaf links geretourneerd. Het algoritme werkt door voor $A[i]$ te tellen hoeveel array-elementen kleiner zijn dan $A[i]$. Het algoritme is als volgt.

```
(1)   $i := 1$ ; gevonden := False;
(2)  while  $i < n$  and not gevonden do
(3)       $j := 1$ ; kleiner := 0;
(4)      while  $j \leq n$  and kleiner = 0 do
(5)          if  $A[j] < A[i]$  then
(6)              kleiner := kleiner + 1;
(7)          fi
(8)           $j := j + 1$ ;
(9)      od
(10)     if kleiner = 0 then
(11)         gevonden := True;
(12)     fi
(13)      $i := i + 1$ ;
(14) od
(15) if gevonden then
(16)     return  $i - 1$ ;
(17) else
(18)     return  $n$ ;
```

Het *vergelijken* van array-elementen in regel (5) (binnenste while-loop) is maatgevend voor de complexiteit. Dit ga je in **c.** ook aantonen.

a. (5 punten)

Hoeveel arrayvergelijkingen doet het algoritme voor het rijtje 3, 2, 3, 4, 1? En voor 3, 2, 3, 1, 1? Geef voor elke ronde $i = 1$ t/m 4 het aantal arrayvergelijkingen dat wordt gedaan in die ronde plus de waarde van j wanneer de binnenste while stopt.

b. (6 punten)

Leg uit waarom het algoritme voor elk invoerarray ten minste n arrayvergelijkingen doet. Onderscheid hierbij twee situaties: (α) de kleinste waarde komt voor op een positie $< n$, en (β) de kleinste waarde is uniek en staat achteraan op positie n . Hint: wat weet je in situatie (β) van gevonden?

c. (10 punten)

Toon aan dat het *vergelijken* van array-elementen (test in regel (5)) maatgevend is voor de complexiteit van het algoritme. Doe dit door het aantal keren dat een regel/operatie wordt gedaan in verband te brengen met (d.w.z. af te schatten op) het aantal keren dat deze test wordt uitgevoerd. Laat zo zien dat elke andere operatie hooguit even vaak als (een constante keer) de test in regel (5) plaatsvindt. De dubbele test in regel (2) mag je als één test beschouwen. Hetzelfde geldt voor regel (4).

d. (6 punten)

Neem aan dat we te maken hebben met situatie (α) , dus de kleinste waarde uit A komt voor op een positie $< n$.

(i) Op een array waarin $A[1]$ de kleinste waarde is doet het array het minimale aantal arrayvergelijkingen, namelijk n . Leg uit waarom dit in situatie (α) de enige invoer is waarvoor de ondergrens uit **b.** bereikt wordt, door te laten zien dat voor elke andere invoer echt meer dan n arrayvergelijkingen nodig zijn.

(ii) We bekijken nu een speciaal soort invoer. Laat het eerste voorkomen van de kleinste waarde op plek $1 < i < n$ staan en veronderstel dat verder geldt dat $A[i] < A[i-1] < A[i-2] < \dots < A[2] < A[1]$. Hoeveel arrayvergelijkingen doet het algoritme voor deze invoer? Je antwoord moet afhangen van i en n . Voorbeeld met $n = 8$: 4, 3, 2, 1, 5, 3, 1, 2.

Vanaf nu nemen we aan dat we te maken hebben met situatie (β) , dus we weten al dat de kleinste waarde van A uniek is en op positie n staat. We bekijken nu het beste geval en het slechtste geval dat kan voorkomen in deze situatie.

e. (12 punten)

(i) Leid voor $n > 2$ af uit het algoritme waaraan het invoerarray zou moeten voldoen opdat het algoritme op die invoer n arrayvergelijkingen doet. Gebruik dat om te laten zien dat n arrayvergelijkingen niet haalbaar is.

(ii) Leg op basis van het algoritme uit hoeveel arrayvergelijkingen er in het beste geval in situatie (β) worden gedaan en voor wat voor invoerrijtjes A dat voorkomt. Geef een algemene beschrijving van *alle* gevallen, licht je antwoord toe en geef een voorbeeldrijtje met $n = 8$.

Hint: stel dat de op een na kleinste waarde van A op positie $1 \leq k < n$ staat. Hierin is k de eerste plek vanaf links waar die waarde voorkomt. Wat is dan de meeste gunstige k en hoe vaak komt de waarde $A[k]$ voor in het array om het aantal arrayvergelijkingen zo klein mogelijk te krijgen? Leid daaruit en uit het algoritme het gevraagde af.

f. (11 punten)

Leid uit het algoritme af hoeveel arrayvergelijkingen er gedaan worden in het slechtste geval en voor wat voor invoerrijtjes A dat voorkomt¹. Geef *alle* gevallen en leg duidelijk uit hoe je aan je antwoord komt.

Geef ook een illustratief voorbeeld van een worst case invoerrijtje met $n = 8$.

¹Wat moet —in termen van het algoritme— gelden opdat het worst case aantal vergelijkingen wordt bereikt en wat betekent dat voor A ?