Eerste huiswerkopgave Complexiteit 2020
Inleveren: maandag 2 maart 2020
Inleveren bij: `l.t.vinkhuijzen@liacs.leidenuniv.nl`
Maak de uitwerking in LaTeX!

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

Er zijn **39** punten te behalen in deze opgave.

We beschouwen hetvolgende probleem:
Input: Een getal k, een array gehele getallen $A[1], A[2], \ldots, A[n]$.
Output: De index i van een stijgende deelrij van opeenvolgende elementen van A, van lengte k. Dus een i waarvoor geldt dat $A[i] \leq A[i+1] \leq \cdots \leq A[i+k-1]$. Als A niet zo'n deelrij bevat, dan moet de index -1 geretourneerd worden.

Het algoritme hieronder retourneert een index i waar zo'n deelrijtje begint, dus waarvoor geldt dat $A[i] \leq A[i+1] \leq \cdots \leq A[i+k-1]$. Als zo'n deelrij niet bestaat wordt -1 geretourneerd. Het algoritme kijkt voor elke i of er vanaf positie i zo'n deelrij bestaat. Als dat het geval is wordt de beginindex (i dus) bewaard en verder gegaan met de volgende i. Het algoritme gaat als volgt:

```
(1)   i := 1; index := −1;
(2)   while i ≤ n − k + 1 do
(3)       j := i; teller := 1; fout := false;
(4)       while j < i + k − 1 and not fout do
(5)           if A[j] > A[j + 1] then
(6)               fout := true;
              else
(7)               teller := teller + 1;
(8)               if teller = k then
                  // stijgende deelrij van lengte k gevonden
(9)                   index := i;
                  fi
              fi
(10)          j := j + 1;
          od
(11)      i := i + 1;
      od
(12)  return index;
```

a. (10 punten)
Toon aan dat het *vergelijken* van array-elementen (test in regel (5)) maatgevend is voor de complexiteit van dit algoritme. Doe dit door het aantal keren dat een regel wordt gedaan in verband te brengen met het aantal keren dat de test in regel (5) wordt uitgevoerd. Laat hierbij onder andere zien dat de test in regel (5) ten minste $n - k + 1$ keer gebeurt.

b. (7 punten)
Hoeveel arrayvergelijkingen worden er gedaan in het beste geval voor algemene n en k met $k > 2$? En voor wat voor invoerrijtjes komt dat voor? Geef *alle* gevallen en leg op basis van het algoritme uit hoe je aan je antwoord komt.
c. (10 punten)
Hoeveel arrayvergelijkingen worden er gedaan in het slechtste geval voor algemene n en k met $k > 2$? En voor wat voor invoerrijtjes A komt dat voor? Leid dit af uit het algoritme en beschrijf *alle* gevallen.
Tip: Het kan handig zijn om eerst even naar een speciaal geval te kijken om een idee te krijgen van de oplossing. Neem bijvoorbeeld $k = 4$.

We passen het algoritme zo aan dat het meteen stopt zodra in regel (8) een stijgend rijtje ter lengte k is gevonden. In de rest van de opgave bekijken we het aldus aangepaste algoritme.

d. (6 punten)
Hoeveel arrayvergelijkingen worden er gedaan in het beste geval voor algemene n en k met $k > 2$? En voor wat voor invoerrijtjes komt dat voor? Geef *alle* gevallen en leg op basis van het algoritme uit hoe je aan je antwoord komt. Maak onderscheid tussen $k < \frac{n}{2} + 1$, $k > \frac{n}{2} + 1$ en $k = \frac{n}{2} + 1$.
e. (2 punten)
Laat zien dat het aantal arrayvergelijkingen in het slechtste geval zoals bij **c** gevonden, nu niet meer gehaald kan worden.
f. (4 punten)
Stel dat A een stijgend rijtje bevat ter lengte precies ℓ, met $\ell < k$, bijvoorbeeld startend op positie 1. Dus: $A[1] \leq A[2] \leq \cdots \leq A[\ell]$, maar $A[\ell] > A[\ell + 1]$. Hoeveel vergelijkingen van array-elementen doet het aangepaste algoritme dan voor $i = 1, 2, \ldots, \ell$ samen?