

# Dynamisch nakijken

## Programmeeropdracht 3 Algoritmen en Complexiteit 2022

### 1 Probleembeschrijving

Multiple-choice vragen zijn makkelijk na te kijken. Je kent gewoon alleen punten toe als de vraag volledig correct beantwoord is. Lastiger om eerlijk na te kijken zijn echter de vragen waar je een correcte volgorde aan moet geven (bijvoorbeeld in welke volgorde bepaalde historische events hebben plaatsgevonden). Een mogelijke manier om dit na te kijken is bijvoorbeeld het volgende:

Zet het correcte rijtje en het geantwoorde rijtje onder elkaar en geef 1 punt voor elk getal dat op dezelfde plaats staat. Zo krijgt onderstaande inzending bijvoorbeeld score 2.

correct: 254631  
gegeven: 251463

Dit is echter niet de meest eerlijke manier om na te kijken. De bovengegeven inzending zou namelijk hierdoor maar 2 van de 6 punten krijgen, terwijl het rijtje vrijwel correct is, alleen de 1 staat op de verkeerde plek. Een betere manier om na te kijken is door niet naar de posities te kijken, maar in plaats daarvan te kijken naar hoeveel getallen je minimaal moet wegstrepen om een correct rijtje te verkrijgen. Voor elk getal dat niet weggestreept wordt, krijg je 1 punt. Aan het voorbeeld hierboven zou nu score 5 toegekend worden, aangezien slechts 1 getal weggestreept hoeft te worden om de rijtjes gelijk te maken.

correct: 254631  
gegeven: 251463

Jij bent de aangewezen persoon om een flink aantal van deze toetsen na te kijken. Jij, als echte programmeur, hebt natuurlijk geen zin om dit allemaal handmatig na te moeten kijken en wil dit werk graag automatiseren. Gelukkig zijn alle antwoorden van de studenten al digitaal voor jou aangeleverd. Het enige dat je nog hoeft te doen is een programma schrijven om aan alle antwoorden een score toe te kennen op de wegstreepmethode.

## 2 Invoer/uitvoer

Als invoer staan op de eerste regel zowel het aantal inzendingen  $k$  die je moet nakijken als het correcte rijtje (alle rijtjes bestaan uit de getallen 0-9). Vervolgens staan op de volgende  $k$  regels per regel het rijtje van één inzending. Als uitvoer wordt voor elke inzending de score verwacht die dat rijtje krijgt gevolgd door hoe het overgebleven correcte rijtje zonder de weggestreepte getallen eruitziet. In het geval er meerdere rijtjes van dezelfde lengte mogelijk zijn, geef dan het lexicografisch eerste rijtje (dus het rijtje dat in getalvorm het kleinst is).

### 2.1 Voorbeeldinvoer

```
4 143625
123456
143625
432561
543621
```

### 2.2 Voorbeelduitvoer

```
3 125
6 143625
4 4325
4 4362
```

## 3 Inleveren

**Controleer goed of jouw Pythonversie Python 2 of Python 3 is en zorg dat je Python 3 gebruikt. Zorg er in ieder geval dat je altijd haakjes zet om je print statements (dus gebruik `print("hello world")` in plaats van `print "hello world"`). Dit is een van de dingen waar bij de vorige nakijksessie een aantal programma's op stuk liepen.**

**Let op:** Deze programmeeropdracht wordt voor een groot deel automatisch nagekeken. Zorg dus dat je je precies aan onderstaande richtlijnen houdt, anders kan je code niet worden nagekeken.

### Controleer of je code goed werkt voor het inleveren

Door `check.sh` uit te voeren in dezelfde map als je python- en input/output- files kun je controleren of je code de juiste uitvoer geeft op de kleine input (`small.input`) en het juiste inputformaat heeft.

### Nakijkeisen

- De code dient geschreven te zijn in Python 3.
- De code leest de input van *stdin* en schrijft naar *stdout*. Om eenvoudig invoer te testen kun je gebruikmaken van input redirection:  
`python ex3.py < small.input`

*Hint: de volgende voorbeeldcode leest op de juiste manier van *stdin* en schrijft hetzelfde naar *stdout*.*

```
import fileinput
for line in fileinput.input():
    # line bevat al een newline van zichzelf
    print(line, end="")
```

- Het codebestand zelf moet de naam `ex3.py` hebben.
- Lever **alleen** het python bestand met je code in en geef het de naam `ex3.py`. Lever dus geen zip/tar in of iets dergelijks.
- Je uitwerking dient binnen 10 seconden antwoord te kunnen geven op zowel de large als small input.

### Hoe wordt de code nagekeken?

Er zijn twee inputfiles aan de hand waarvan je code wordt nagekeken:

- Bijgeleverd is `small.input` met de bijbehorende output `small.output`. Als je code dezelfde output geeft met `small.input` als input, wordt in principe het cijfer 6 toegekend.  
*NB: Om deze score te behalen is het wel noodzakelijk dat je daadwerkelijk redelijkerwijs het algoritme hebt geïmplementeerd. Simpelweg altijd dezelfde output retourneren wordt niet goedgekeurd.*
- Daarnaast is ook voor elke student apart (in `large.input.zip`) een inputfile `large_{studentnummer}.input` bijgeleverd (zonder de bijbehorende output). Als je code ook de juiste output voor deze input geeft, wordt in principe het cijfer 10 toegekend.

De code wordt hiernaast gecontroleerd op plagiaat van anderen en code op het internet. Hierop is de algemene plagiaat- en frauderegeling van de UvA van toepassing.

### Geen inputfile ontvangen / overige vragen

Mocht je studentnummer niet tussen de inputfiles staan of mocht je overige vragen/problemen hebben over het inputformaat, stel een vraag tijdens het werkcol-

lege of stel je vraag op Canvas / via email.