

# Algoritmisch denken

- [2011 januari examen](#)
- [2012 januari examen](#)
- [2014 januari examen](#)
- [2015 januari examen](#)
- [2017 januari examen](#)
- [2019 juni examen](#)
- [2020 augustus examen](#)
- [2020 januari examen](#)

# 2011 januari examen

- Je kreeg een robotprog code en 3 verschillende plattegronden. Je moest de weg van het robotje + zijn eindplaats en richting aangeven. Ook werden er sommige vakjes "gemarkeerd".
- Van een code kunnen afleiden wat voor soort algoritme het is (Bubblesort, Quicksort,...). De code was dan nog niet 'optimaal' en er moesten nog kleine verbeteringen aan worden gebracht. Ook moest je hierbij de complexiteit uitleggen ervan.
- Zelf een programma schrijven (Je had 25 leerlingen, 6 vakken, elk vak onthoud het grootste cijfer en hoe vaak deze persoon al dit vak had gehad. De leerkracht moest simpel een punt kunnen ingeven voor een vak en jouw programma moest dit dan wegschrijven in een array. Het 2de programma moest uit de array's lezen en een output tonen.

Alternatieve versie:

- Vrij eenvoudige pseudocode om twee-dimensionaal array te bewerken. Lees de vraag goed na, er zaten addertjes in. - Rij van cijfers analyseren, kijken welk sorteeralgoritme het is, en dan het algoritme uitschrijven in pseudocode. - Robotprog: code volgen en voor drie plattegronden zeggen waar de robot staat, in welke richting, of hij crasht en op hoeveel een teller staat.

# 2012 januari examen

*1ste vraag:* je hebt een computerlokaal met maximaal 27 pcs, die hebben allemaal een nummer. Je moet 4 methoden schrijven: -voegPcToe -verwijderPc -isPcAanwezig -hoeveelPcsAanwezig

*2de vraag:* je krijgt een ongesorteerde rij van 5 elementen en je moet daar bubble sort op toepassen. Je krijgt daar een tabel bij waar je stap per stap moet tonen welke elementen je verwisselt. je moet dan op de gesorteerde rij binair zoeken toepassen

Andere mogelijkheid:

Je moet een briefje trekken met daar de naam van één sorteeralgoritme op. Die moet je dan mondeling uitleggen aan de hand van een aantal speelkaarten. Je moet ook de complexiteit van het sorteeralgoritme kunnen geven.

*3de vraag:* schrijf mbv het stappenplan dat als bijlage zit de volgende methode. Doe enkel de eerste 4 stappen van het stappenplan

*4de vraag:* test volgende methode: een methode die het gemiddelde berekent. Die klopt helemaal niet, er staat bijvoorbeeld "for(int i=0;i<=rij.length;i++){ " en die <= is dus fout. Dan moet je die fouten ook nog verbeteren

## Met mevrouw Lens

*1ste vraag*

Je krijgt de volgende opgave 'Je moet een programma maken voor een klant. De bedoeling is dat je een boodschappenlijstje maakt.' Leg uit hoe je dit maakt en maak de java functie om producten aan dit lijstje toe te voegen. Vervolgens krijg je een array van goederen met de winkel waar ze het goedkoopst zijn. Schrijf de java code van hoe je een boodschappenlijstje kan genereren per winkel. De functie moet als parameters de arraylist van opgave 1 en de naam van de winkel krijgen. De retourwaarde is de gemaakte arraylist.

*2de vraag*

Je krijgt een array. Leg uit wat bucket sort doet, stap voor stap en geef de verschillende stappen in

de array. Vervolgens, leg uit: binair zoeken. In hoeveel stappen vind met het gevraagde getal en pas dit weer toe op de array.

### *3de vraag*

Zelfde als hierboven.

### *4de vraag*

Je krijgt een volledige functie, leg stap voor stap uit wat deze doet en geef per stap een paar testgetallen zodat de leerkracht kan zien dat je deze begrijpt.

# 2014 januari examen

## Marina Lens

### 1. Mondeling

Een bepaald zoek/sort algorithm uitleggen:

- selection sort + lineair/binair zoeken
- insertion sort
- bubble sort + optimisaties
- merge sort
- quick sort

--> Je krijgt een aantal speelkaarten, deze gebruik je om te tonen hoe het sortalgorithm werkt. Dan moet je vragen over de code beantwoorden (code wordt NIET gegeven), en uiteindelijk iets kort zeggen over de complexiteit van dit Algorithm (bvb. C van Bubble Sort =  $n^2$ , waarom? omdat je een for lus binnen een andere for lus hebt - dat is voldoende uitleg voor Mvr. Lens).

### 2. Schriftelijk

#### 1. Datastructuren

Vraag: Je moet een foto album aanmaken. Een foto album bestaat uit max. 50 mappen, met in elke map max. 250 foto's. Je moet van iedere map de naam bijhouden, bvb. "Barcelona". Je kunt geen 2 of meer mappen hebben met dezelfde naam. Voor elke foto moet je de naam (bvb. "Barcelona Strand 1") bijhouden, en de grootte ervan in KB. Ook hierbij mogen er geen 2 foto's zijn met dezelfde naam.

--> Teken de datastructuren en declareer ze in Javascript.

Mogelijke Oplossing:

```
var namen = new Array(50);
var inhouden = new Array(50);
for (var i = 0; i < inhouden.length; i++){
  .....inhouden[i] = new Array(250);
  .....for(var j = 0; j < inhouden[i].length; j++){
    .....inhouden[i][j] = new Array(2);
  .....}
}
var mappen = [namen,inhouden]
```

## 2. Functies

A. Vraag: Schrijf de functie om, gegeven de naam (bvb. "strand") en grootte (bvb. "56" //kb) van een foto en de naam van een map, deze foto toe te voegen aan deze map. Indien deze map niet bestaat moet deze worden aangemaakt. Deze functie moet true of false returnen (true = gelukt, false = niet gelukt). Schrijf alle functionaliteiten hiervoor op.

Mogelijke Strategie?

```
function vindMapIndex(mapNaam){
// gegeven de naam van een map, return index; als deze map niet bestaat, return index -1}
function vindLegeFotoIndex(mapIndex){
// gegeven het index van een map, return de eerste index in de map waar nog geen foto staat;
indien de map vol is (= geen lege indexen), return -1}
function voegMapToe(mapNaam){
//gegeven mapNaam en indien vindMapIndex(mapNaam)==-1 (dus map bestaat nog niet), maak
nieuwe map op eerst volgende lege plaats (gemakkelijkst is om deze lege plaats te vinden door in
de rij van namen te kijken waar eerste "undefined" staat)
//returned index van deze nieuwe map}
function voegFotoToe(mapNaam,fotoNaam,aantalKB){ //hoofdfunctie voor deze vraag
...var toegevoegd = false;
...var mapIndex = vindMapIndex(mapNaam);
...if (mapIndex==-1 /*map bestaat niet*/) mapIndex=voegMapToe(mapNaam); //maakt nieuwe
map en bewaar index
...if(mapIndex!=-1){
.....var fotoIndex = vindLegeFotoIndex(mapIndex);
.....if(fotoIndex!=-1){
.....inhouden[mapIndex][fotoIndex][0] = fotoNaam;
.....inhouden[mapIndex][fotoIndex][1] = aantalKB;}
.....toegevoegd = true; //foto is toegevoegd
.....}
...}
...return toegevoegd;
}
```

B. Vraag: Maak een functie die een rij terug geeft met daarin de naam van iedere map en de totale grootte van ieder van deze mappen. Simpel? Neen, want er is een "catch": De rij moet gesorteerd zijn op grootte van map, van kleinste naar grootste. Dus eerste map in deze rij moet de kleinste zijn, 2de de 2de kleinste, etc.. Maak alle functionaliteiten die hierbij nodig zijn.

Mogelijke strategie?

1. Een functie maken die een nieuwe rij returned met daarin de grootte van map 1, map 2, map 3, ... map 50.

De indexen hiervan moeten hetzelfde zijn als de indexen van de namen van de mappen.

Vb. nieuweRij = [505,550,612,65]

Als je namen rij ["Barcelona","Madrid","Spanje","Strand"] is, dan komt 505 overeen met de map "Barcelona", 550 met "Madrid" etc..

2. Maak een aangepaste sort functie (bvb. bubbleSort) die twee rijen als parameters mee neemt. Dit is eenvoudiger dan het klint. De sort vergelijkt alleen de groottes enz. van de grootte rij, het enig dat je moet bijvoegen is dat wanneer dit algorithm een rij hersorteerd (dus rij1[i] naar rij1[j] beweegt bvb.), dat hij ook de namen rij hersorteerd. Uiteindelijk krijg je de namen hersorteerd volgens de grootte van iedere map (deze grootte rij is dan natuurlijk ook gesortd).

3. Maak de hoofdfunctie voor deze vraag die een 2D rij maakt. Eerste D is de namenrij (die gesorteerd wordt met ons aangepaste sort algorithm) en de tweede D is de rij met groottes van deze mappen (die ook gesorteerd is door ons sort algorithm). LET OP: Deze nieuwe rij moet maar evenlang zijn als het aantal mappen die in gebruik zijn, dus niet alle 50 max. aantal mappen (var aantalMappen = vindLegeMapIndex() //aparte functie die gewoon in namen rij kijkt en de eerste lege index ('undefined') teruggeeft).

C. Vraag: Maak een functie die, gegeven n en m, een rij terug geeft met daarin een collage van  $n*m$  foto's. Deze collage bestaat uit willekeurige foto's uit willekeurige mappen. Wel bevat het geen 2 dezelfde foto's. Deze collage is een rij die bestaat uit strings. Een string moet er bvb. als volgt uitzien, "Barcelona\Beach1.jpg". Maak alle functionaliteiten die hiervoor nodig zijn.

Mogelijke strategie?

Deze vraag valt goed mee:

```
function checkString(string,array){
...var stringFound = false;
...for (var i = 0; i < array.length && stringFound == false; i++){
.....if(array[i]==string) stringFound = true;
...}
...return stringFound;
}
function collage(n,m){
...var uitArray = new Array(n*m);
...for (var i = 0; i < uitArray.length; i++){
.....var aantalMappen = (vindLegeMapIndex());
.....var mapIndex = parseInt(Math.random()*aantalMappen);
.....var mapString = namen[mapIndex];
.....var aantalFotos = (vindLegeFotoIndex(mapIndex));
.....var fotoIndex = parseInt(Math.random()*aantalFotos);
.....var fotoString = inhouden[mapIndex][fotoIndex][0]; //0 is waar naam zich bevindt
.....var finalString = mapString + "\" + fotoString + ".jpg"; //"\" ipv "\"?
.....if(checkString(finalString,uitArray)==false){ //zorg dat deze foto niet al in collage staat
.....uitArray[i] = finalString;
.....}
.....else{ i--; }
...}
...return uitArray;
```

}

# Greetje Jongen

## Vraag 1a:

Schrijf een functie die een markt genereert, deze markt heeft rijen en in elke rij een identiek aantal kraampjes.

Elk kraampje heeft een categorie, kunst, brol, food. Niet alle plaatsen zijn bezet.

## Vraag 1b:

Schrijf een functie die in een string de totale aantallen van alle categorieën teruggeeft, dus bv: "Aantal kraampjes met brol: 22, aantal kraampjes met food: 33, aantal kraampjes met kunst: 25, aantal onbezette kraampjes: 21".

## Vraag 2:

De organisator wil dingen fair houden en dus geen kraampjes van dezelfde categorie op de plaatsen N - Z - O - W van een gegeven kraampje, schrijf een functie die dit controleert voor een gegeven categorie en true teruggeeft indien de plaatsing correct is.

## Vraag 3:

De koning komt langs op de markt, deze wandeld niet graag en stapt dus van de linkerbovenhoek naar rechts,

dan van rechts naar links op de volgende rij, de volgende links naar rechts en zo verder. Hij wil geen lege/onbezette kraampjes tegenkomen.

Schrijf een functie die een voorgestelde markt (als 2D array) meekrijgt en deze herorganiseert zodat alle lege kraampjes in de laatste rijen staan.

(Wat niet gezegd werd op het examen was dat er dan geen rekening moest gehouden worden met de "Faire" positionering van de kraampjes, dus mochten kraampjes van dezelfde categorie WEL naast elkaar staan)

# 2015 januari examen

(Lector: F. Vogels)

Er waren 5 vragen (op de PC):

- Maak een grid aan
  - `function createGrid(width, height, value){`
  - `var grid = new Array(width);`
  - `for(var i = 0; i < grid.length; i++){`
  - `grid[i] = new Array(height);`
  - `for(var j = 0; j < grid[i].length; j++){`
  - `grid[i][j] = init;`
  - `}`
  - `}`
  - `return grid;`
  - `}`
- Pas de kleurencombinatie aan van het grid
- roteer het grid 180°
- /
- Recursievraag
  - Je kreeg een getal als parameter en elk cijfer van dat getal moest je -1 doen
    - Vb: 459 => 348
    - Indien je cijfer 0 was dan werd dat cijfer een 9

# 2017 januari examen

De eerste oefeningen waren allemaal gebaseerd op een spel 4-op-een-rij. 'V' stelt een leeg veld voor, 'G' een gele schijf en 'R' een rode schijf. Deze werden in een 2-dimensionale array geplaatst.

1. Maak een rooster. Maak een rooster waarvan de afmetingen (rijen en kolommen) worden meegegeven. Het rooster is volledig leeg.

Voorbeeld:

aantalRijen: 3, aantalKolommen: 5

Verwacht resultaat:

V V V V V

V V V V V

V V V V V

2. Controleer op zwevende schijven.

Controleert of er ergens rode of gele schijven staan waar nog een lege plaats onder is. De functie geeft true terug als er geen zwevende schijven zijn.

Voorbeeld:

V V V V V

V G V G R

G R V V R

Hier is een zwevende schijf, namelijk die op kolom 4, waar de 'G' boven een leeg vak 'V' hangt.

3. Zoeken naar leeg veld.

Zoekt naar het eerste lege veld in een bepaalde kolom, van onder naar boven te zoeken.

Voorbeeld:

V V V V R

G V V G R

R V G R G

Het eerste lege element op kolom 2 is 3 (cel op de onderste rij). Het eerste lege element op kolom 4 is 1.

#### 4. Roteren.

Schrijf een functie om het raster 90° met de klok mee te draaien.

Voorbeeld:

```
V V R V R      R G V
G G R V G  wordt R G V
R R G G G      G R R
              G V V
              G G R
```

#### 5. Recursieve vraag

Een visboer heeft een aquarium met een bepaalde capaciteit. Op week 0 is dit volledig vol. Iedere week verkoopt hij 60 vissen, en komen er 5% bij. Schrijf een recursieve functie om voor een bepaalde week het aantal vissen te berekenen.

# 2019 juni examen

Examen door Van Hee:

Dit examen was met P5.js en bestond uit 3 opdrachten.

## 1. Fotobewerking

Je krijgt een foto en moet het eerste deel ervan rood kleuren, het tweede deel groen en het derde deel blauw.

## 2. Zwevende letters

Laat het programma in het begin 100 random letters aanmaken, deze zweven constant rond met een bepaalde snelheid als ze de rand raken veranderen ze van richting. Als de letters 10px van elkaar zijn dan moeten ze samen gevoerd worden tot 1 woord en zweven ze verder. Laat telkens ook nieuwe letters bijmaken als er een letter samengevoegd wordt of weg gaat.

## 3. ????

# 2020 augustus examen

Het examen van Augustus 2020 met dank aan ISW: Examen Algo augustus 2020.zip

P5 met Van Hee (Niet meer van toepassing) met dank aan ISW en Beddegenoodts Elias:

# 2020 januari examen

Het examen van Januari 2020 met dank aan ISW: Examen Algo Januari 2020.zip

P5 met Van Hee (Niet meer van toepassing) met dank aan ISW: