

TESTED: ONE JUDGE TO RULE THEM ALL

Niko Strijbol

Studentennummer: 01404620

Promotoren: prof. dr. Peter Dawyndt, dr. ir. Bart Mesuere
Begeleiding: Charlotte Van Petegem

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de informatica

Academiejaar: 2019 – 2020



1 INLEIDING

Programmeren -> steeds belangrijker en nuttigere kennis om te hebben Goed programmeren -> vereist veel oefening Zeker in cursussen met meer mensen -> goede ondersteuning lesgever Automatiseren van beoordeling programmeeroefeningen -> Dodona

DIT is de inleiding van mijn thesis. Hallo aan iedereen! Omdat we met een initiaal werken, is het aangewezen dat de eerste alinea redelijk wat tekst bevat.

Verder is dit aan de orde:

$$y = 5 + 6$$

HALLO IK BEN KLEINKAPITAAL!

Hallo, dit is tekst met code: `dit is dan de code.`

ZEEKOMKOMMERS (Holothuroidea) vormen een groep van ongewervelde dieren die behoren tot de klasse van stekelhuidigen. De meeste soorten hebben een langwerpig en worstvormig lichaam dat zowel aan de voor- als achterzijde stomp eindigt. Er zijn ook vormen met een sliertige of een bolle lichaamsvorm.

Zeekomkommers zijn net als andere stekelhuidigen uitgesproken zeedieren; ze kunnen niet overleven in zoet water of op het land. De verschillende soorten komen voor van de getijdenzone tot in de diepzee. De meeste soorten leven op de bodem, waar ze zich kruipend voortbewegen. Ook zijn er strikt gravende soorten bekend en enkele soorten zijn zelfs goede zwemmers.

De meeste groepen van stekelhuidigen hebben een stervormig lichaam (radiaal symmetrisch). Zeekomkommers wijken hier uitwendig sterk van af. De inwendige anatomie vertoont echter wel grote gelijkenissen met andere stekelhuidigen zoals zeesterren, slangsterren en zee-egels. Er zijn ongeveer 1.700 verschillende soorten zeekomkommers beschreven, waarvan er slechts twee weleens langs de kust van België en Nederland worden gevonden.

Zeekomkommers leven van kleine, zwevende voedingsdeeltjes of kleine organismen die uit het water worden gefilterd. Ze hebben hiertoe vaak duidelijke tentakels rond hun mond waarmee ze hun voedsel verzamelen. Sommige vertegenwoordigers zijn carnivoor en eten kleine diertjes, andere leven enkel van organische deeltjes in de modder van de zeebodem. De dieren worden zelf gegeten door verschillende andere dieren zoals roofvissen en zeeschildpadden. Sommige soorten kunnen een bepaald deel van het darmstelsel uitstoten ter verdediging. Deze sliertige structuren zijn plakkerig en bovendien giftig.

Verschiedende soorten worden vooral in Azië gebruikt in gerechten of worden gebruikt als een traditioneel medicijn. Zeekomkommers worden vaak eerst gedroogd voor ze worden verwerkt als voedsel of in medicijnen.

Zeekomkommers lijken in het geheel niet op de andere stekelhuidigen maar hebben inwendig een sterk vergelijkbare bouw. Een van de belangrijkste verschillen is het feit dat de zeekomkommer een voor- en een achterzijde heeft. Feitelijk ligt een zeekomkommer altijd op zijn zijkant. Bij de

zee-egels en de zeesterren hebben de dieren een boven- en onderzijde. De anus is bij deze groepen aan de bovenzijde gelegen en de mondopening aan de onderzijde.

Rond de slokdarm is in het lichaam van de zeekomkommer een harde, ringvormige structuur aanwezig die bestaat uit sterk verkalkt weefsel. Deze ring heeft niet zozeer een verstevigende functie zoals botten van hogere dieren maar dient als aanhechtingspunt voor verschillende lichaamsspieren. Kalkhoudende structuren komen binnen de stekelhuidigen veel voor maar hebben meestal slechts een beschermende functie. De vorm van de kalkring verschilt per groep, soorten uit de familie Paracucumidae hebben een 'kale' kalkring zonder aanhangsels, die uit de familie Sclerodactylidae hebben enkelvoudige aanhangsels aan de kalkring en die uit de familie Placothuriidae bezitten lange en gepaarde aanhangsels.

Zeekomkommers hebben van alle stekelhuidigen het hoogst ontwikkelde vaatstelsel, ze gebruiken lichaamsvloeistof als bloedvloeistof en voor het transport van voedingsstoffen en gasen door het lichaam. Het zijn de enige stekelhuidigen die een met meerdere kamers gevulde lichaamsholte hebben die sterk lijkt op het hart van hogere gewervelde dieren. De endeldarm van sommige soorten bevat een speciaal orgaan dat dient ter verdediging. Dit orgaan ontspruit aan de waterlongen en wordt wel aangeduid met de buizen van Cuvier, zie ook onder het kopje vijanden en verdediging.

De vijfvoudig straalsgewijze lichaamsbouw aan de buitenzijde van het lichaam komt bij de meeste soorten duidelijk tot uiting in het watervaatstelsel. Dit is een stelsel van kanalen, blaasjes en lichaamsuitstulpingen. Het watervaatstelsel ontspruit uit het ringvormige, verkalkte kanaal rondom de slokdarm, het ringkanaal, en bestaat uit vijf radiaalkanalen. Ieder radiaalkanaal staat in verbinding met een van de vijf rijen kleine zakvormige uitstulpingen aan de buitenzijde die vaak -maar niet altijd- op een evenredige afstand van elkaar gelegen zijn. Deze uitstulpingen worden de voetjes (podia) genoemd.

Bij de meeste stekelhuidigen, zoals zeesterren en zee-egels, staat het watervaatstelsel in verbinding met het omringende zeewater. Deze verbinding wordt het steenkanaal genoemd omdat de wanden van het kanaal verkalkt zijn. De eigenlijke verbinding is de madreporiet of zeefplaat, dit is een verhard en verkalkt plaatje met kleine poriën waardoor zeewater het lichaam in wordt gezogen. Zeekomkommers hebben tot wel 100 steenkanalen, maar deze eindigen vrijwel altijd binnen in de lichaamsholte en staan niet in verbinding met de huid. De dieren gebruiken dus geen zeewater als vloeistof in het watervaatstelsel zoals veel andere stekelhuidigen, maar lichaamsvocht. Alleen van soorten uit de orde Elapipoda zijn uitwendige madreporieten bekend.

De functie van de inwendig gelegen steenkanalen en zeefplaten is niet geheel duidelijk. Waarschijnlijk dienen de structuren om de waterhuishouding van het lichaamsvocht en bloedvloeistof te reguleren. Uit het ringkanaal ontspruiten ook verschillende blaasjes van Poli. Deze dienen waarschijnlijk eveneens om het watervaatstelsel te ondersteunen als opslagvat. Uiterlijk lijken de blaasjes op de ampullen die aan de voetjes gebonden zijn, maar ze zijn groter en in het lichaam gelegen. De blaasjes van Poli zijn niet direct verbonden met de radiaalkanalen zoals bij de ampullen het geval is. Het aantal blaasjes kan variëren van een enkel blaasje tot meer dan vijftig.

Sommige zeekomkommers ademen met behulp van hun tentakels. Dit zijn in feite omgebouwde buisvoetjes die hun oorspronkelijke functie hebben verloren. Alle andere soorten ademen met behulp van waterlongen. Deze hebben eenzelfde functie als de kieuwen van vissen. Waterlongen komen alleen voor bij zeekomkommers en zijn niet bekend bij andere dieren, ook niet bij andere stekelhuidigen.

De waterlongen zijn sterk vertakte structuren die ontspruiten aan de achterzijde van het lichaam, ze staan in verbinding met de endeldarm. De waterlongen zijn in feite uitlopers van de cloaca die

zich ontwikkeld hebben tot ademhalingsorganen. Ze vullen een groot deel van de lichaamsholte en lopen door tot vooraan in het lichaam. De zeekomkommer haalt adem door de waterlongen vol water te zuigen via de anus en de gespierde endeldarm. In de waterlongen vindt de gasuitwisseling plaats waarbij zuurstof wordt opgenomen en afvalstoffen worden afgegeven. Sommige zeekomkommers zuigen meerdere keren per minuut zeewater in en uit en anderen 'ademen' heel diep in om vervolgens het water er in één keer weer uit te persen. Hierbij wordt zuurstof afgegeven aan de lichaamsvloeistof die door het lichaam stroomt. Vele kleine trilharen die de binnenzijde van de lichaamsholte bekleden zorgen dat de stroming van de lichaamsvloeistof de juiste kant op gaat

Zeekomkommers hebben geen hart of een vergelijkbaar orgaan en ze hebben ook geen echt bloed. Wel stroomt er een vloeistof door het lichaam die afgescheiden is van het watervaatstelsel en voedingsstoffen naar de lichaamscellen brengt. Deze kleurloze vloeistof wordt de bloedvloeistof genoemd en bestaat uit lichaamsvocht dat verrijkt is met zuurstof, afkomstig van het ademhalingsapparaat. De ruimtes in het lichaam waar bloedvloeistof aanwezig is worden de bloedlacunes genoemd en het geheel van kanalen wordt met bloedlacunestelsel aangeduid. De verbindingskanalen tussen de bloedlacunes lijken op de aderen van hogere dieren, maar hebben geen eigen vaatwand. Het zijn feitelijk kanaalachtige ruimtes tussen de verschillende lichaamsweefsels waarin de bloedvloeistof stroomt.

Door samentrekkingen van de lengtespieren van de darm wordt de bloedvloeistof in het lichaam rondgepompt. De darmwand is voorzien van vele zeer fijn vertakte bloedlacunes die voedingsstoffen onttrekken uit de darm. De voedingsstoffen worden door vijf kanalen verdeeld over het lichaam, deze kanalen zijn onder de radiaalkanalen van het watervaatstelsel gelegen. De uiteinden van deze bloedlacunes zijn eveneens vertakt maar ze eindigen doodlopend zodat het geheel geen ringvormig, gesloten systeem is.

„Dit is duidelijk voor iedereen”.

2 DODONA

2.1 Wat is Dodona?

Intro over Dodona: korte geschiedenis, terminologie, hoe Dodona werkt (oefeningen, judges, enz.)
-> Over de judge wordt in het deel hierna meer verteld.

2.2 Evalueren van een oplossing

Zoals reeds vermeld worden de oplossingen van studenten geëvalueerd door een zelfgeschreven evaluatiescript, de *judge*. In wezen is dit een eenvoudig programma: het krijgt de configuratie via de standaardinvoerstream (stdin) en schrijft de resultaten van de evaluatie naar de standaarduitvoerstream (stdout). Zowel de invoer als de uitvoer van de judge zijn json. Het formaat van deze uitwisseling ligt vast in een json-schema, dat publiekelijk beschikbaar is. ¹

Een judge ondersteunt één programmeertaal. In de praktijk ondersteunt elke judge oplossingen in de programmeertaal waarin hij geschreven is, m.a.w. de Java-judge ondersteunt Java, de Python-judge Python, enz. Ook heeft elke judge een eigen manier waarop de testen voor een oplossing opgesteld moeten worden. Zo worden in de Java-judge junit-testen gebruikt, terwijl de Python-judge doctests en een eigen formaat ondersteunt.

In grote lijnen verloopt het evalueren van een oplossing van een student als volgt:

1. De student dient de oplossing in via de webinterface van Dodona.
2. Dodona start een Docker-image met de judge.
3. De judge wordt uitgevoerd, met als invoer de configuratie, zoals hierboven vermeld.
4. De judge evalueert de oefening aan de hand van de code van de student en de evaluatiecode opgesteld door de lesgever (ie. de junit-test, de doctests, ...).
5. De judge vertaalt het resultaat van deze evaluatie naar het Dodona-formaat en schrijft het uit.
6. Dodona vangt die uitvoer op, en toont het resultaat aan de student.

2.3 Probleemstelling

Het huidige systeem waarop de judges werken resulteert in twee nadelen. Bij het bespreken van de nadelen is het nuttig een voorbeeld in het achterhoofd te houden, teneinde de nadelen te kunnen concretiseren. Als voorbeeld gebruiken we de „Lotto”-oefening², met volgende opgave:

¹Een tekstuele beschrijving is te vinden in de handleiding (Dodona-team 2020).

²Vrij naar een oefening van prof. Dawyndt.

De **lotto** is een vorm van loterij die voornamelijk bekend is vanwege de genummerde balletjes, waarvan er een aantal getrokken worden. Deelnemers mogen zelf hun eigen nummers aankruisen op een lottoformulier. Hoe groter het aantal overeenkomstige nummers tussen het formulier en de getrokken balletjes, hoe groter de geldprijs.

Opgave

Schrijf een functie `loterij` waarmee een lottotrekking kan gesimuleerd worden. De functie moet twee parameters `aantal` en `maximum` hebben. Aan de parameter `aantal` kan doorgegeven worden hoeveel balletjes a er moeten getrokken worden (standaardwaarde 6). Aan de parameter `maximum` kan doorgegeven worden uit hoeveel balletjes m er moet getrokken worden (standaardwaarde 42). Beide parameters kunnen ook weggelaten worden, waarbij dan de standaardwaarde gebruikt moet worden. De balletjes zijn daarbij dus genummerd van 1 tot en met m . Je mag ervan uitgaan dat $a \leq m$. De functie moet een string teruggeven die een strikt stijgende lijst van a natuurlijke getallen beschrijft, waarbij de getallen van elkaar gescheiden zijn door een spatie, een koppelteken (-) en nog een spatie. Voor elk getal n moet gelden dat $1 \leq n \leq m$.

Oplossingen voor deze opgave staan in codefragmenten 2.1 en 2.2, voor respectievelijk Python en Java.

Het belangrijkste nadeel aan de huidige werking is het bijkomende werk voor lesgevers indien zij hun oefeningen in meerdere talen willen aanbieden. De Lotto-oefening heeft een eenvoudige opgave en oplossing. Bovendien zijn de verschillen tussen de versie in Python en Java minimaal, zij het dat de Java-versie wat langer is. Deze oefening zou zonder problemen in nog vele andere programmeertalen geïmplementeerd kunnen worden.

Ook bij ingewikkeldere oefeningen die zich concentreren op algoritmen, waar de uiteindelijke taal van de implementatie niet relevant is. Een voorbeeld hiervan is het vak „Algoritmen en Datastructuren” dat gegeven wordt door prof. Fack aan de wiskunde³. Daar zijn de meeste opgaven vandaag al beschikbaar in Java en Python op Dodona, maar dan als afzonderlijke oefeningen.

Het evalueren van een oplossing voor de Lotto-oefening is minder eenvoudig, daar er met willekeurige getallen gewerkt wordt: het volstaat niet om de uitvoer gegenereerd door de oplossing te vergelijken met een op voorhand vastgelegde verwachte uitvoer. De geproduceerde uitvoer zal moeten gecontroleerd worden met code, specifiek gericht op deze oefening, die de verwachte vereisten van de oplossing controleert. Deze evaluatiecode moet momenteel voor elke programmeertaal en dus elke judge opnieuw geschreven worden. In de context van ons voorbeeld controleert deze code bijvoorbeeld of de gegeven getallen binnen het bereik liggen en of ze gesorteerd zijn.

Voor de lesgevers is het opnieuw opstellen van deze evaluatiecode veel en repetitief werk. Duur het twee minuten om deze code te schrijven en een vak heeft 30 oefeningen, dan duurt het een uur. In twee talen duurt al twee uur, tien talen vraagt al tien uur.

Het tweede nadeel aan de huidige werking, maar wel veel kleiner van belang, betreft het implementeren van de judges zelf. Hoewel de interface voor de judges eenvoudig is, blijkt in de praktijk dat het implementeren van een judge verre van eenvoudig is. Uiteraard is een deel van die complexiteit ingevolge het evalueren van de code, een taak die niet in alle talen eenvoudig is. Doch is

³De studiefiche is voor de geïnteresseerden beschikbaar op <https://studiegids.ugent.be/2019/NL/studiefiches/C002794.pdf>

```

import java.util.HashSet;
import java.util.Set;
import java.util.concurrent.ThreadLocalRandom;
import java.util.stream.Collectors;

class Main {

    public static String loterij(int aantal, int maximum) {
        var r = ThreadLocalRandom.current();
        var result = new HashSet<Integer>();
        while (result.size() < aantal) {
            result.add(r.nextInt(1, maximum + 1));
        }
        return result.stream()
            .sorted()
            .map(Object::toString)
            .collect(Collectors.joining(" - "));
    }

    public static String loterij(int aantal) {
        return loterij(aantal, 42);
    }

    public static String loterij() {
        return loterij(6, 42);
    }
}

```

Codefragment 2.1: Voorbeeldoplossing in Java.

```

from random import randint

def loterij(aantal=6, maximum=42):
    getallen = set()
    while len(getallen) < aantal:
        getallen.add(randint(1, maximum))

    return " - ".join(str(x) for x in sorted(getallen))

```

Codefragment 2.2: Voorbeeldoplossing in Python.

de complexiteit ten dele te wijten aan de noodzaak om in elke judge opnieuw de hele evaluatieprocedure te implementeren. Aan het deel van de code dat het resultaat van een evaluatie omzet naar het Dodona-formaat is niets taalspecifiek.

Het probleem hierboven beschreven laat zich samenvatten als volgende onderzoeksvraag, waarop deze thesis een antwoord wil bieden:

Is het mogelijk om een judge zo te implementeren, dat de opgave en evaluatiecode van een oefening slechts eenmaal opgesteld dienen te worden, waarna de oefening beschikbaar is in alle talen die de judge ondersteunt?

Enkele zaken zijn expliciet buiten de focus van deze thesis gehouden, zoals:

1. of de nieuwe judge het eenvoudiger maakt om nieuwe talen toe te voegen in vergelijking met het maken van een volledig nieuwe judge,
2. de *front end* van Dodona, waaronder
 - a) het automatisch aanpassen van codefragmenten in een opgave naar de taal de van oplossing,
 - b) het kiezen van een taal van indienen.

2.4 Opbouw

Het volgende hoofdstuk van deze thesis handelt over het antwoord op bovenstaande vraag. Daarna volgt ter illustratie van het gebruik van de judge een gedetailleerde beschrijving van hoe een nieuwe taal moet toegevoegd worden aan de thesis. Dat hoofdstuk heeft ook ten doel te dienen als documentatie voor zij die de judge willen gebruiken. Tot slot wordt afgesloten met een hoofdstuk over beperkingen van de huidige implementaties, en waar er verbeteringen mogelijk zijn (het „toekomstige werk”).

Hallo

Questionable tastes are not provided in this document. fidelity Thank you! IJzer, ijzer Goossens,
Mittelbach en Samarin [1993](#).

INDEX

judge, [4](#)

BIBLIOGRAFIE

Dodona-team (23 oktober 2020). *Creating a new Judge*. Universiteit Gent. URL: <https://dodona-edu.github.io/en/guides/creating-a-judge/>.

Goossens, Michel, Frank Mittelbach en Alexander Samarin (1993). *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley.