

# Taak: Algoritmen en Datastructuren 1

Sam Van den Vonder ([sam.van.den.vonder@vub.be](mailto:sam.van.den.vonder@vub.be))  
Youri Coppens ([youri.coppens@vub.be](mailto:youri.coppens@vub.be))

27 november 2023

Upload je oplossing vóór **maandag 4 december 2023 om 13:00 uur** via [Canvas](#). Upload één .zip bestand dat dezelfde structuur volgt als de template code die gepubliceerd is op Canvas, dus een zip bestand met daarin een folder **taak**, met daarin de bestanden **samenvatting.rkt** en **taak.rkt** (jouw implementatie). De deadline is strikt. Laattijdige inzendingen worden ongeacht de omstandigheden niet geaccepteerd.

In deze opdracht implementeer je een variatie van het brute force pattern matching algoritme uit de cursus. Eerst leggen we uit hoe deze variatie werkt (sectie 1), daarna geven we concrete instructies over hoe je aan deze opdracht begint (sectie 2), en onderaan krijg je belangrijke informatie i.v.m. plagiaat (sectie 3).

## 1 Een variatie op brute force

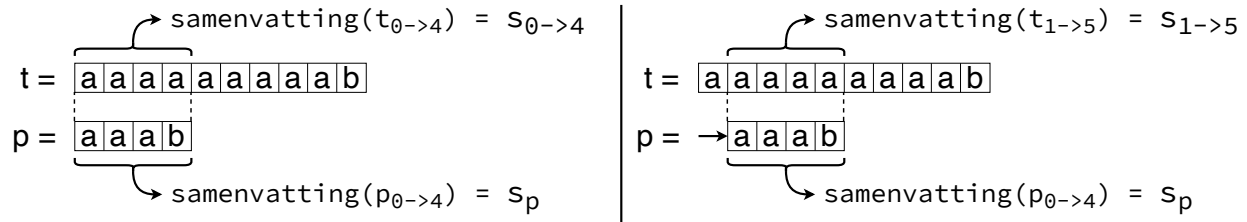
Bij brute force is het probleem dat er in het slechtste geval voor elke alignment van een patroon en tekst pas een mismatch is helemaal achteraan het patroon. Bijvoorbeeld, in figuur 1 wordt het patroon "aaab" vergeleken met de tekst "aaaaaaaaab". Dit patroon zal voor elke alignment pas een mismatch geven op het karakter b in het patroon, behalve op het einde van de tekst waar het patroon zal matchen. In deze opdracht willen we het brute force algoritme verbeteren door een potentiële mismatch sneller uit te sluiten.

We kunnen veel van de mismatches zoals die uit figuur 1 al aan het begin vermijden door, op elke alignment van tekst en patroon, eerst een ruwe *samenvatting* van het patroon te vergelijken met een ruwe *samenvatting* van een stukje van de tekst met dezelfde lengte als het patroon. Dit is afgebeeld in figuur 2 (links): Gegeven een tekst  $t$  en een patroon  $p$ , vergelijken we eerst een samenvatting van de tekst  $s_{0 \rightarrow 4}$  met een samenvatting van het patroon  $s_p$ . De notatie  $t_{0 \rightarrow 4}$  en  $s_{0 \rightarrow 4}$  betekent dat we enkel kijken naar de tekst  $t$  op interval  $[0, 4[$  (dus van index 0 tot index 4), en dat de samenvatting berekend werd van enkel deze substring.

- Indien de samenvatting van het patroon **niet** overeenkomt met de samenvatting van de tekst ( $s_{0 \rightarrow 4} \neq s_p$ ), dan is er sowieso een mismatch en kan het algoritme verder met de volgende

t =	a	a	a	a	a	a	a	a	a	b
	:	:	:	:	:	:	:	:	:	:
p =	a	a	a	b						

Figuur 1: Een patroon dat bij brute force pas mismatcht op het laatste karakter.



Figuur 2: Een eerste potentiële match van het algoritme (links), en een volgende potentiële match na een eerste mismatch (rechts).

alignment van tekst en patroon om opnieuw te proberen, dus zoals afgebeeld in figuur 2 (rechts). Hier wordt opnieuw eerst een samenvatting van de tekst vergeleken met het patroon, maar deze keer voor de karakters in de tekst van index 1 tot 5.

- Indien de samenvatting van het patroon **wél** overeenkomt met de samenvatting van de tekst ( $s_{0 \rightarrow 4} = s_p$ ), dan is er **mogelijk** een match tussen de tekst en het patroon. Het algoritme kan deze match bevestigen op dezelfde manier als brute force door de karakters van het patroon en de tekst met elkaar te vergelijken (op de huidige alignment van patroon en tekst). Let op dat hier dus nog steeds een mismatch kan ontstaan, in welk geval het algoritme verder gaat met de volgende alignment van het patroon en de tekst.

Bijgevoegd bij deze opdracht voorzien we een R7RS library die een procedure exporteert genaamd *maak-samenvatting* om van een string een samenvatting te maken, waar elke samenvatting zal worden voorgesteld als een getal. Deze procedure heeft het volgende proceduretype:

(string number number  $\rightarrow$  number)

De procedure **maak-samenvatting** neemt dus 3 argumenten, namelijk een string en 2 indices **a** en **b**. Het resultaat is een samenvatting (een getal) voor de substring van de tekst op het interval  $[a, b[$ . Enkele voorbeelden:

1. Gegeven een patroon "aaab", maak een samenvatting van het volledige patroon (dus van index 0 tot 4)

---

```
> (maak-samenvatting "aaab" 0 4)
821838164759851843
```

---

2. Gegeven een tekst "aaaaaaaaab", maak een samenvatting van de eerste 4 karakters "aaaa" (dus van index 0 tot 4)

---

```
> (maak-samenvatting "aaaaaaaaab" 0 4)
821838164759854963
```

---

3. Gegeven een tekst "aaaaaaaaab", maak een samenvatting van de laatste 4 karakters "aaab" (dus van index 6 tot 10)

---

```
> (define t "aaaaaaaaab")
> (maak-samenvatting t 6 (string-length t))
821838164759851843
```

---

## 2 Opdracht

Op Canvas vind je een bestand `opdracht.zip` met skelet code die je gebruikt als vertrekpunt voor je implementatie. Hierin vind je een folder `taak` met 2 files:

**samenvatting.rkt:** Dit bestand is een R7RS library die de procedure `maak-samenvatting` implementeert en exporteert. In dit bestand maak je geen wijzigingen.

**taak.rkt:** Dit bestand is een R7RS library die je gebruikt om de opdracht te implementeren. De library in de gegeven code exporteert de `match` procedure van het brute force pattern matching algoritme (dewelke je gaat aanpassen).

Om het bestand `taak.rkt` uit te voeren in DrRacket zal je de folder “taak” moeten installeren als package zoals gezien in het WPO. Kort samengevat ging dit zo: pak de zipfile `opdracht.zip` uit, en plaats de folder `taak` op een locatie naar wens. In DrRacket, navigeer in de menubalk naar “File -> Install Package...”. In deze dialoog, klik op “Browse...” en selecteer “Directory”. Selecteer daarna de `taak` folder die je hebt uitgepakt, en klik op “Install”. Na succesvolle installatie mag je de dialoog sluiten.

**Opdracht** Pas de `match` procedure van `taak.rkt` aan zodat deze de variatie op brute force implementeert zoals beschreven in sectie 1.

**Hint** Tijdens het verloop van het algoritme zijn er meerdere momenten mogelijk om de nodige samenvattingen te berekenen.

- Een mogelijk moment is om de samenvattingen te berekenen terwijl het algoritme over de tekst loopt.
- Een mogelijk moment is om een preprocessing stap te introduceren zoals bij de geziene algoritmes KMP en Quicksearch. In plaats van het patroon te preprocessen (zoals bij KMP en Quicksearch) kan je in dit geval de tekst preprocessen.

## 3 Plagiat

### 3.1 Wat is toegelaten?

De taak dient strikt individueel en op eigen kracht gemaakt te worden. Dat betekent dat je jouw taak op een zelfstandige basis moet maken en je jouw werk moet kunnen uitleggen, onder toezicht moet kunnen herimplementeren, en je werkwijze moet kunnen verdedigen. Werk (code, tekst, etc.) overnemen van of delen met derden (bv. medestudenten, websites, GitHub, etc.) is verboden. Merk op: het is dus ook expliciet verboden om werk gegenereerd door AI tools (bv. gebaseerd op “large language models”) in te dienen als eigen werk. Elektronische hulpmiddelen worden gebruikt om alle inzendingen met online bronnen en met elkaar te vergelijken, zelfs over verschillende academiejaren heen.

- De enige code die mag overgenomen worden is het code skelet dat wordt aangeboden op Canvas, welke je gebruikt als vertrekpunt voor deze taak.
- Je mag met bronvermelding gebruik maken van de code van de cursus Algoritmen en Datastructuren 1.

### **3.2 Wat is plagiaat?**

Elke handeling van een student die afwijkt van de gegeven instructies en niet in overeenstemming is met het examenreglement wordt beschouwd als onregelmatigheid. Plagiaat is eveneens een onregelmatigheid. Onder plagiaat wordt begrepen het gebruik maken van andermans werk, al dan niet in bewerkte vorm, zonder nauwkeurige bronvermelding (cf. OER, Artikel 118§2). Plagiaat kan betrekking hebben op verschillende vormen van werk zoals tekst, code, beeld, etc.

### **3.3 Wat gebeurt er als studenten verdacht worden?**

Elk vermoeden van plagiaat zal onverwijld aan de decaan van de faculteit worden gerapporteerd. Zowel gebruiker als verlener van zulke code zullen worden gerapporteerd en zullen worden behandeld volgens de plagiaatregels van het examenreglement (cf. OER, Artikel 118). De decaan kan beslissen tot (een combinatie van) de examentuchtsancties, gaande van 0 punten op het werkstuk tot een verbod tot (her)inschrijving voor één of meerdere academiejaren (cf. OER, Artikel 118§5).

Contacteer ons als je twijfelt of iets al dan niet als plagiaat zou beschouwd worden.