Practicum 3 Sorting & Searching

Substring search en regular expressions

a. Implementatie van substring in programmeertalen

Ga voor Java en minstens nog één **andere** programmeertaal na hoe het substring algoritme is geïmplementeerd in die taal[1]. Vertel om welke taal het gaat, toon de source code van het algoritme en verklaar om welke variant het gaat (brute force, Knuth-Morris-Pratt, Boyer-Moore, Rabin-Karp of misschien nog wel iets anders).

Let op: de methode waarnaar je op zoek bent hoeft natuurlijk niet substring te heten. In Java heet deze methode bijvoorbeeld `indexOf()`, wat een methode van de `String` class is.

b. Vergelijken van algoritmes

Algoritmes

In deze deelopgave ga je de algoritmes Knuth-Morris-Pratt en Boyer-Moore uit (Sedgewick & Wayne, 2011) met elkaar vergelijken. De code mag je natuurlijk overnemen uit het boek. De code moet enigszins worden aangepast, want we zijn niet geïnteresseerd waar de substring zich bevindt, maar wel hoe vaak deze voorkomt in de tekst.

Woorden

Je moet van tien zelf gekozen woorden bepalen hoe vaak ze in de tekst voorkomen. Neem van alle smaken wat: korte woorden, lange woorden, en neem ook een aantal woorden die niet in de tekst voorkomen. Per woord tel je hoeveel karaktervergelijkingen het algoritme doet. Uiteindelijk krijg je dus een tabel waarin per woord staat hoe vaak het in de tekst voorkomt en voor beide algoritmes het aantal karaktervergelijkingen geeft. Ga na of je resultaten in lijn zijn met de tabel “cost summary for substring search implementations” in (Sedgewick & Wayne, 2011, p. 779) en bespreek dat in je verslag.

Tekst

Neem als tekst waar in je gaat zoeken het beroemde gedicht “Mei” van Herman Gorter, dat uit 4000 verzen bestaat. Het is te vinden op http://cf.hum.uva.nl/dsp/ljc/gorter/mei.boek1.html, http://cf.hum.uva.nl/dsp/ljc/gorter/mei.boek2.html en http://cf.hum.uva.nl/dsp/ljc/gorter/mei.boek3.html.

LET OP: dit is bij elkaar één tekst. Plak de teksten aan elkaar en bewaar dat in één document. Gebruik dit document bij je experimenten.

BELANGRIJK: Het kan zijn dat je bij opdracht 3 (sub-string search) in het gedicht van Herman Gorter een `ArrayIndexOutOfBoundsException` krijgt in de code van KMP en/of BM. Dit kan komen door een fout in Boek 2 van het gedicht. Je kunt dit controleren door te zoeken naar 'hun gladde '.

[1] Hiermee wordt dus bedoeld: hoe hebben de ontwerpers van de taal het zoeken van een substring in een string geïmplementeerd. Hoe ziet de source van de programmeertaal eruit?

Direct daar achter zou moeten staan 'leêeen'. De kans is echter groot dat je iets hebt staan zoals 'le?een' met op de plek van het vraagteken een héél vreemd karakter of een vraagteken in een ruit. Als je dat vreemde karakter vervangt door de e-dakje (ê) dan werkt het zoals verwacht.

(In de HTML staat er op die plek '≖' terwijl dit had moeten zijn 'ê'. Er mist dus een c aan het einde.)

c. Reguliere expressies

Gebruik de java class `java.util.regex.Pattern` om een Java methode

```
boolean checkURL(String url)
```

te maken die controleert of een url voldoet aan de volgende regels.

- De URL begint met http(s) of ftp(s).
- Het top-level domein is of nl of edu.
- Er is minimaal één derde-level domein en als het derde-level domein www is dan is er minimaal een vierde-level domein.
- Het second-level domein is nooit korter dan 3 karakters.
- Tussen iedere slash zit minimaal 2 karakters en die begint nooit met een cijfer.
- Als er parameters voorkomen in de URL dan eindigt de naam van de parameter altijd met een cijfer.

http://lib.hva/nl is een valide URL

ntp://www.hva.nl/a/b?tijd=UTC is geen valide URL.

Maak de regex zo kort mogelijk. Gebruik commentaar en goede naamgeving om de regex helemaal toe te lichten. Maak unittests aan om de methode te testen met diverse testwaarden. Verzin goede testgevallen waarin je zoveel mogelijk randgevallen test. Neem tests die slagen en tests die falen. Beschrijf waarom je juist deze hebt toegevoegd. Toon ook de output van de unit test in je verslag.

Voor meer informatie over top-level en second-level domeinen zie:
https://nl.wikipedia.org/wiki/Topleveldomein en
https://nl.wikipedia.org/wiki/Secondleveldomein .

Geciteerde werken

Sedgewick, R., & Wayne, K. (2011). *Algorithms.* Pearson Education.