Westerdals

Oslo School of Arts,
Communication and Technology

PG4200
Algoritmer og datastrukturer

Innlevering 2

Besvarelsen leveres i It's Learning innen
11. november 2015 klokken 23.55

1 Introduksjon

Her skal vi se på hvordan vi kan lage en enkel søkemotor for web-innhold. De viktigste oppgavene er til en søkemotor er

- Bygge søkeindeks: Tråling og indeksering. Besøke websider og samle informasjon i passende datastrukturer. På engelsk: *web crawling*.
- Søking: Svare på søkeforespørsler.

Praktisk brukbare søkemotorer løser som regel disse oppgavene fortløpende. I vår enkle variant skal vi gjøre dette sekvensielt: Vi starter med å bygge søkeindeksen. Deretter tillates brukeren å søke i den.

1.1 Tråling

Webtråling kan utføres på følgende måte:

- Velg ut en webside.
- Samle relevant informasjon om innholdet på websiden.
- Samle alle lenkene på websiden.
- Gjenta prosedyren med hver og en av lenkene, og fortsett så lenge det passer, f.eks. til du ikke ønsker å samle mer data.

I vår lille søkemotor skal vi registrere *ordene* som forekommer på websidene, d.v.s. tekststrenger som er omgitt av mellomrom.

1.2 Søking

I denne oppgaven går søking ut på følgende:

- Brukeren angir et søkeord.
- Programmet returnerer en liste over websider der søkeordet forekommer.

I denne oppgaven nøyer vi oss med å kunne søke etter *enkeltord*. I mer brukervennlige søkemotorer kan man naturligvis søke etter flere ord om gangen.

1.3 Indeksering

Søkeindeksen er en enkel database over ordforekomster i websider, som bygger bro mellom søkingen og trålingen. Når man lager en slik gjelder det å bruke datastrukturer som sikrer:

- Lav kjøretid i forbindelse med innsamling av data
- Lavt forbruk av minneressurser
- Lav responstid i forbindelse med søk

2 Oppgave

Oppgaven går ut på å skrive en klasse `MyEngine.java` som implementerer grensesnittet `SearchEngine.java` (Beskrevet i seksjon 3).

Med en tilfredsstillende implementasjon av `MyEngine.java`, kan man bruke `SimpleFrontEnd.java` til å søke bruke og teste søkemotoren.

Det finnes mange muligheter for å finpusse en slik implementasjon, og vi har her definert følgende delmål:

Delmål 1: Elementær indeksering av nettinnhold

Det første målet er rett og slett å få `MyEngine.java` til å fungere som en tilfredsstillende implementasjon av `SearchEngine.java`.

Legg merke til at du kan overse `setBreadthFirst` og `setDepthFirst` i denne omgang.

Delmål 2: Avgrensning av indekseringen

Sørg for at kun de ordene som forekommer i `words.txt`, men som ikke forekommer i `stopwords.txt` blir indeksert.

Delmål 3: Kontrollere traverseringsorden

Implementer `setBreadthFirst` og `setDepthFirst` slik at de faktisk kan brukes til å styre traverseringsordenen.

Delmål 4: Effektiv minnebruk

Til slutt: Hvordan kan man bygge en søkeindeks som tar minst mulig minneressurser?

For å bedømme dette blir løsningen testet for minnebruk under følgende forutsetninger:

- Alle dataene skal lagres i datamaskinens minne.
- Vi skal samle inntil 32768 ordforekomster. Det er altså greit om systemet krasjer når vi forsøker å fylle på flere forekomster.
- Vi måler minnebruken med `SimpleFrontEnd.memoryFootprintInMegaBytes`.

Totalvurdering av innleveringen

Karakterskala: A-F

Vekt: 20%

Grupper: 1-2 personer. I vurderingen blir det ikke tatt hensyn til gruppestørrelsen.

Karakterfastsettelse: Karakteren fastsettes utifra kvaliteten på det som leveres og antall delmål som det er arbeidet med. Dersom man kun oppnår det første delmålet, er man sikret ståkarakter en god margin. Dersom man leverer god kvalitet på to av delmålene, vil man være sikret en C.

Praktiske opplysninger

Besvarelsen leveres elektronisk i en `zip`-fil `innlevering2.zip` som skal inneholde

- `MyEngine.java`
- `README.txt`
- Eventuelle andre hjelpeklasser.

3 Om utdelt kode

3.1 `MyEngine.java`

Dette er begynnelsen på en implementasjon av en søkemotor. Oppgaven går ut på å levere en klasse `MyEngine.java` som implementerer grensesnittet `SearchEngine.java`, og det kan være en god idé å ta utgangspunkt i den utleverte `MyEngine.java`.

3.2 `SearchEngine.java`

Dette er grensesnittet som forklarer hva en søkemotor skal gjøre.

3.3 `SimpleFrontEnd.java`

Dette er en enkel tekstbasert front-end som kan brukes til å teste ut søkemotoren.

Legg merke til at `SimpleFrontEnd` er avhengig av at `MyEngine.java` er en tilfredsstillende implementasjon av `SearchEngine`-grensesnittet.

3.4 `WebPageReader.java`

For å gjøre webtrålingen enklere, har faglærer skrevet klassen `WebPageReader`. Lenkene på web-siden får vi tak i ved å kalle metoden `getLinks`. Innholdet får vi tak i ved å kalle metoden `getWords`. Vi nøyer oss altså med å registrere hvilke ord som brukes på websiden. Selve indekseringen av en enkelt side startes ved metodekallet `run()`.

For et eksempel på hvordan man kan bruke `WebPageReader` til å skaffe seg materiale fra en webside, se `WebPageReader.main`.

Legg merke til at `WebPageReader` ganske sikkert har et betydelig forbedringspotensiale. Det er likevel meningen at den skal være god nok for bruk i denne innleveringen.