



INFORMASJONSVITENSKAP

INFO 232, Høst 2014

Andre Obligatoriske Oppgave

Utlevert 31. oktober 2014

Innleveringsfrist 20. november 2014, klokken 18:00

1 Regler

Oppgaven vurderes til bestått eller ikke bestått.

Oppgaven leveres individuelt.

Leveringsformatet er enten en enkelt tekstfil med programmet, (enten som .pl eller .prolog) eller et arkiv.

Arkivet skal være enten zip, tgz eller tar.gz. *Rar godtas ikke*. Merk at prolog ikke har en standardisert måte å håndtere tegn som ikke er i ASCII. (Tenk på til dømes Æ, Ø og Å.) Derfor bør dere ikke bruke dem.

2 Om datafilene

2.1 reiseruter.txt

reiseruter.txt inneholder kvintuple linjer med termer som beskriver en reiserute:

- Første linje er stedet som en reiser fra.
- Andre linje er stedet en reiser til.
- Tredje linje er reisemåten (buss, tog, bat, fly).
- Fjerde linje er tiden som brukes i minutter.
- Femte linje er kostnaden i norske kroner.

For eksempel vil linjene:

'Odense' .

'Aalborg' .

'tog' .

182 .

361 .

bety at det går et tog fra Odense til Aalborg, turen tar 3 timer og 2 minutter (182 minutter), og koster 361 kroner.

2.2 stedsinfo.txt

- Første linje er navnet på byen.
- Andre linje er folketall.
- Tredje linje er status (storby (> 30 000 innbyggere), småby (< 30 000 innbyggere), hovedstad)
- Fjerde linje er en funfact
- Femte linje er x-koordinatet (antall kilometer øst for et nullpunkt)¹

¹Siden Hastur er et islandsk reisebyrå, har vi valgt kilometer sør og øst for Reykjavik.

- Sjette linjer er y-koordinatet (antall kilometer nord for et nullpunkt)

For eksempel:

```
'Kristianstad'.
```

```
'35711'.
```

```
'storby'.
```

```
'Sveriges laveste punkt ligger her, 2,41 meter under havet.'.
```

```
3361.
```

```
849.
```

3 Om koden

Dere får utlevert to kodebiter. En begynnelse på A-star algoritmen, og et predikat som leser en fil med termer, og returnerer en liste av termer. Merk at predikatet som returnerer en liste med termer inneholder alle termene i tekstfilen, men også en term på slutten 'end_of_file'.

4 Oppgaven

4.1 Del 1: Les og tolk data

Hastur (Når det ekki berre hastur, det hastur, hastur, hastur!) er er islandsk turselskap for folk i farten. De har tenkt å lansere en webapplikasjon som finner veien mellom forskjellige byer i Skandinavia, og ønsker å gi den ut i Norge først. Dere skal lage en liten prototype på en slik applikasjon. Oppgaven er delt opp i 2 deler. Lesing og tolking av data, og stifinning.

Dataene ligger i fillageret, og har navnene reiseruter.txt, og stedsinfo.txt.

4.1.1 Oppgave 1.1

Skriv predikatet `beskriv/1`. `beskriv/1` tar inn et bynavn og skriver ut en kort beskrivelse av den. Følger vi eksempelet over skal vi skrive ut:

```
beskriv('Kristianstad').
```

```
Kristianstad er en storby med 35711 innbyggere.
```

```
Sveriges laveste punkt ligger her, 2,41 meter under havet.
```

```
true.
```

predikatet skal alltid holde gitt at det finnes en by med det oppgitte navnet.

4.1.2 Oppgave 1.2

Skriv predikatet `listByer/1`. `listByer/1` gir en liste over alle byer (kun bynavn).

4.1.3 Oppgave 1.3

Skriv predikatet `rute/2`. `rute/2` skal returnere sann hvis og bare hvis det finnes en direkte rute mellom to punkter. For eksempel skal `rute('Odense', 'Aalborg')` holde, fordi det finnes en togrute mellom disse to stedene. Hvis det ikke finnes en slik rute, skal predikatet ikke holde.

Disse predikatene skal altså kunne kjøres etter at filene ble innlest.

4.2 Del 2: Stifinning

Dere skal nå lage stifinneren. Dere skal finne en rute mellom to gitte steder i listen av steder dere har fått oppgitt.

Hovedpredikatet dere skal skrive er `finnReiseRaskest/2`. `finnReiseRaskest/2` tar inn to steder, `Fra` og `Til`, og skal skrive ut en liste over den raskeste måten å komme seg dit på, og gi en samlet pris. Gitt spørringen: `Fra = 'Bergen', Til = 'Bardufoss'`², `finnReiseRaskest(Fra, Til)`.

Kan en mulig utskrift være:

Fly fra Bergen til Oslo.

Fly fra Oslo til Bardufoss.

Pris: 2321 kroner.

true.

Dere står fritt til å gjøre utskriften penere/bedre etter eget skjønn, men dere må ta med alle stegene i reisen. For å finne ruten skal dere bruke A*-algoritmen beskrevet i boken (se kapittel 12).

Dere får en A*-algoritme i fillageret dere kan bruke, eller dere kan bruke deres egen.

For at algoritmen dere får utdelt skal virke, må dere skrive tre predikater: `goal`, `s` og `h`.

goal er en test på om dere er i mål eller ikke.

s er en suksessorfunksjon som velger neste node som skal besøkes.

h er en heuristikk som sier noe om hvor nærme de forskjellige nodene er målet.

Heuristikken dere velger å bruke må være admissible. Et godt utgangspunkt vil være euklidisk distanse (som er det samme som luftlinje). Et vanlig passasjerfly har en marsjhastighet på om lag 800 km/h, og er dermed alltid det raskeste alternativet. Nevn i en kommentar hvorfor heuristikken din er admissible. (Hvilke krav stilles en heuristikk, og oppfyller din heuristikk disse kravene?)

²Bardufoss er ikke med i datasettet deres