Oblig 2 IN2010

Sophus Bredesen Gullbekk & Mathias Svoren

Kjøring av programmet:

Python standard pakker

- *csv*
- *collections*
- *heapq*

Nødvendige ekstra pakker

- *requests*
- *beautifulsoup*

Installer de ekstra pakkene med

```
$ pip install requests
```

```
$ pip install beatifulsoup
```

kjør programmet med

```
$ python3 oblig2.py
```

Oppgave 1: Bygg grafen

Vi bygger en graf basert på et datasett fra IMDB. Datasettet består av to TSV-filer, `movies.tsv` og `actors.tsv`. Grafen representerer datasettet slik at hver node er en skuespiller og to skuespillere har en kant mellom seg for hver film de har spilt sammen.

Vi leser filene i funksjonen **readfile(movies_filename, actors_filename)**. Grafen konstrueres i funksjonen **buildgraph(actors_in_movie, movies_and_rating, actor_and_movies)**.

Vi representerer grafen som en tuppel med tre elementer. Det første elementet **V** representerer nodene og er et sett som inneholder alle skuespillerne. Det andre elementet **E** representerer kantene og er en *defaultdict* som vil initialisere hver nøkkel som et tomt sett. Det tredje elementet **w** er en *dictionary* som inneholder den minste vekten til alle kantene (kalkulert via ratingen til filmene). For å spare tid og minne, legger vi kun til den minste vekten mellom to skuespillere fordi dette er alt vi trenger for å løse oppgavene.

For å konstruere grafen løper vi gjennom alle skuespillerne. For hver skuespiller løper vi gjennom alle filmene skuespilleren spiller i og for hver av disse filmene løper vi gjennom alle de andre skuespillerne i filmen.

Hvis vi definerer $|V|$ som antall noder, $|E|$ som antall kanter og $|F|$ som antall filmer, så vil kjøretiden til algoritmen være $\mathcal{O}(|V||E||F|) = \mathcal{O}(|V||E|)$ siden $|F|$ kan regnes som en konstant.

Etter å ha kjørt **oblig2.py** skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 1

Nodes: 108733
Edges: 4760823
```

Oppgave 2: Six Degrees of IMDB

De to funksjonene **bfs_shortest_paths_from(G,s)** og **bfs_shortest_path_between(G,s,e)** bruker *bredde først søk* til å finne den korteste stien som forbinner to noder. **bfs_shortest_paths_from(G,s)** er en standard bredde først algoritme som returnerer en *dictionary* med foreldrenodene til rot-skuespilleren (**s**). Denne algoritmen har en verst mulig kjøretid på $\mathbf{O(|E| + |V|)}$. **bfs_shortest_path_between(G,s,e)** vil returnere den opptimale stien mellom de to skuespillerne, verste kjøretid er $\mathcal{O}(|V|)$. Den samlede kjøretiden til opgave 2 er dermed $\mathcal{O}(|E| + |V|)$.

Etter å ha kjørt **oblig2.py** skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 2

Donald Glover
===[ Lennon or McCartney 5.3 ] ===> Kevin Spacey
===[ Margin Call 7.1 ] ===> Jeremy Irons


Scarlett Johansson
===[ Avengers: Endgame 8.4 ] ===> Robert Downey Jr.
===[ A Century of Cinema 5.7 ] ===> Denzel Washington


Carrie Coon
===[ Avengers: Infinity War 8.4 ] ===> Chris Hemsworth
===[ Avengers: Age of Ultron 7.3 ] ===> Julie Delpy


Christian Bale
===[ Empire of the Sun 7.7 ] ===> Burt Kwouk
===[ Beyond Borders 6.5 ] ===> Angelina Jolie


Atle Antonsen
===[ In Order of Disappearance 7.2 ] ===> David Sakurai
===[ Acts of Vengeance 5.7 ] ===> Paz Vega
===[ Kill the Messenger 6.9 ] ===> Michael K. Williams
```

Oppgave 3: Chilleste vei

Vi har vektet grafen med ratingen til filmene som representerer kantene. Vi vil nå finne den chilleste stien: en sti mellom to skuespillere som går gjennom de beste filmene. Vektfunksjonen vi bruker er 10 - rating, slik at den chilleste veien vil tilsvare den korteste i den vektede grafen.

Vi bruker *Dijkstra's* algoritme til å finne den korteste (eller chilleste) veien. Funksjonen **dijkstra(G,s)** implementerer Dijkstra's algoritme. Den går gjennom alle nodene i grafen og finner den korteste stien fra rot-skuespilleren til alle andre skuespillere i settet. Funksjonen vil returnere en dictionary med alle stiene, som senere tolkes av funksjonen **chillest_path_between(G,s,e)**. Vi ender opp med en

liste som inneholder den chilleste stien, samt den samlede vekten til stien. Den samlede kjøretiden til oppgave 3 blir $\mathcal{O}(|E| + |V| * log|V|)$.

Etter å ha kjørt **oblig2.py** skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 3

Donald Glover
===[ The Martian 8.0 ] ===> Enzo Cilenti
===[ The Man Who Knew Infinity 7.2 ] ===> Jeremy Irons
Total weight: 4.8

Scarlett Johansson
===[ Avengers: Infinity War 8.4 ] ===> Josh Brolin
===[ American Gangster 7.8 ] ===> Denzel Washington
Total weight: 3.8

Carrie Coon
===[ Avengers: Infinity War 8.4 ] ===> Samuel L. Jackson
===[ Avengers: Age of Ultron 7.3 ] ===> Julie Delpy
Total weight: 4.3

Christian Bale
===[ The Dark Knight Rises 8.4 ] ===> Liam Neeson
===[ For the Love of Spock 7.6 ] ===> Angelina Jolie
Total weight: 4.0

Atle Antonsen
===[ In Order of Disappearance 7.2 ] ===> Stellan Skarsgård
===[ Good Will Hunting 8.3 ] ===> Casey Affleck
===[ Gone Baby Gone 7.6 ] ===> Michael K. Williams
Total weight: 6.9
```

Oppgave 4: Komponenter

Algoritmen for **components(G)**, går gjennom skuepillerne i grafen og gjør bfs fra skuespillerne om de ikke er i visited. For hver bfs blir traverserte noder lagt til i visited.

I verste tilfelle er kjøretidskompleksiteten $\mathcal{O}(|V| * (|V| + |E|))$, men de fleste skuespillerne er i en stor komponent med 103 099 skuespillere, så antall skuespillere i visited blir fort mange og i realiteten er det antall komponenter * (skuespillere + kanter).

Etter å ha kjørt oblig2.py skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 4

There are 1 components of size 103099
There are 1 components of size 19
There are 1 components of size 10
There are 3 components of size 9
There are 1 components of size 8
There are 5 components of size 7
There are 8 components of size 6
There are 14 components of size 5
```

```
There are 40 components of size 4
There are 113 components of size 3
There are 297 components of size 2
There are 4324 components of size 1
```

Oppgave 5: Quote

Denne funksjonen implementerer egentlig ikke en algoritme, men skraper imdb.com etter informasjon. Funskjonen **getMovieQuote(movie_id)** leser en tt-id og printer ut et sitat fra filmen (hvis det ligger sitater ute på IMDB). Vi testet funksjonen på filmene *Borat, Goldfinger, The Dark Knight* og *The Big Lebowski*.

Etter å ha kjørt oblig2.py skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 5: Quote

Quote from Borat:
=================
Borat: You telling me the man who try to put a rubber fist in my anus was a homosexual?
=================


Quote from Goldfinger:
======================
James Bond: Do you expect me to talk?
Auric Goldfinger: No, Mr. Bond, I expect you to die!
======================


Quote from The Dark Knight:
===========================
The Chechen: [During a private sit down meeting with the gangsters] What do you propose?
The Joker: It's simple. We, uh, kill the Batman.
mobsters laugh
Salvatore Maroni: If it's so simple, why haven't you done it already?
The Joker: If you're good at something, never do it for free.
===========================


Quote from The Big Lebowski:
============================
The Dude: [repeated line by The Dude and others] That rug really tied the room together.
============================
```

Oppgave 5: Least sexistic path

Funksjonen **create_actress_dict(infile)**, leser en tsv fil med skuespillere, om de er actor eller actress og hvilke filmer de spiller i. Den lager en dictionary med antall actresses i hver film og totale antall skuespillere i hver film. Denne funksjonen leser bare en fil, så kjøretidskompleksiteten vil bare avhenge av størrelsen på tsv-filen. I denne oppgaven brukte vi en IMDB datafil med over 11 millioner linjer, så den tok mye lenger tid enn å lese filene i oppgave 1.

Bortsett fra at den gjør litt mindre arbeid for hver loop, så har funksjonen **women_weights()** samme kjøretidskompleksitet som **buildgraph()**, altså $\mathcal{O}(|V||E||F|)$.

least_sexistic_path(G, w_w, s, e) finner beste veien mellom to noder. Den bruker dijkstra til å finne alle de beste veiene fra en node 's' og finner også alle foreldrenodene til alle nodene. Så returnerer den en liste med nodene man må gjennom for å nå 'e'. Funksjonen **dijkstra_women(G, w_w, s)** er dijkstra algoritmen bare på et annet dictionary med vekter. Så den samlede kjøretidskompleksiteten er fortsatt $\mathcal{O}(|E| + |V|log|V|)$.

PS: For å kjøre **women_weights()**, **least_sexistic_path(G, w_w, s, e)** og **dijkstra_women(G, w_w, s)** trenger man tilgang til filen *data.tsv* som er levert i Devilry.

Etter å ha kjørt oblig2.py skal blant annet følgende output bli gitt i terminalen:

```
Oppgave 5: Least sexistic path

Donald Glover
===[ The To Do List (actresses: 1) ] ===> Mae Whitman
===[ Going Shopping (actresses: 10) ] ===> Bruce Davison
===[ Camp Hope (actresses: 2) ] ===> Andrew McCarthy
===[ Jump (actresses: 1) ] ===> Harvey Fierstein
===[ Bullets Over Broadway (actresses: 2) ] ===> Jennifer Tilly
===[ Deal (actresses: 1) ] ===> Burt Reynolds
===[ Big City Blues (actresses: 1) ] ===> William Forsythe
===[ Halloween (actresses: 3) ] ===> Malcolm McDowell
===[ Inhabited (actresses: 1) ] ===> Rosalind Chao
===[ The Man from Elysian Fields (actresses: 1) ] ===> James Coburn
===[ Bite the Bullet (actresses: 1) ] ===> Ian Bannen
===[ Damage (actresses: 2) ] ===> Jeremy Irons
Total acresses: 26

Scarlett Johansson
===[ Rough Night (actresses: 1) ] ===> Demi Moore
===[ The Seventh Sign (actresses: 5) ] ===> Jürgen Prochnow
===[ The Fall (actresses: 2) ] ===> Craig Sheffer
===[ Instant Karma (actresses: 2) ] ===> Larry B. Scott
===[ Wilma (actresses: 3) ] ===> Denzel Washington
Total acresses: 13

Christian Bale
===[ Knight of Cups (actresses: 6) ] ===> Natalie Portman
===[ Anywhere but Here (actresses: 2) ] ===> Caroline Aaron
===[ Anna (actresses: 1) ] ===> Sally Kirkland
===[ Bite the Bullet (actresses: 1) ] ===> James Coburn
===[ The Man from Elysian Fields (actresses: 1) ] ===> Rosalind Chao
===[ Inhabited (actresses: 1) ] ===> Malcolm McDowell
===[ Halloween (actresses: 3) ] ===> William Forsythe
===[ Big City Blues (actresses: 1) ] ===> Burt Reynolds
===[ Deal (actresses: 1) ] ===> Jennifer Tilly
===[ Bullets Over Broadway (actresses: 2) ] ===> Harvey Fierstein
===[ Dr. Jekyll and Ms. Hyde (actresses: 1) ] ===> Polly Bergen
===[ How To Pick Up Girls (actresses: 1) ] ===> Abe Vigoda
===[ Love Is All There Is (actresses: 3) ] ===> Angelina Jolie
Total acresses: 24
```