IN2010, høsten 2023

«Obligatorisk» oppgave 3

Grafer

Oppgave 1:

Jeg kjører litt annen skuring her fordi jeg er sær og liker å være vanskelig, men jobber ut fra at noder er skuespillere og kanter er filmer, jeg synes det ga mest mening. Men – tre ting:

1) Jeg legger ikke til noder (skuespillere) som bare har spilt i filmer vi ikke har data på.
2) Jeg legger ikke til kanter (filmer) vi ikke har data på mellom noder (skuespillere) selv om de teknisk sett har spilt sammen. «But this is outside the scope of this paper and left as a task for the reader» eller noe.
3) Alle skuespillernoder har x antall kanter med seg selv for alle filmene de har spilt i. Dette er for å håndtere særtilfeller hvor vi har skuespillere som har spilt i en film vi har data på, men ikke på andre skuespillere som har spilt i den.

Med alt dette i bakhodet får jeg noe færre noder og en del flere kanter enn det oppgaven foreslår jeg skulle fått. Fjern kommentarer fra de tre nederste linjene fra EvenLessHappyLittleGraphs.py og få:

```
$ python EvenLessHappyLittleGraphs.py

Vertices:  124030
Edges:  11460348
It took 27.32 seconds to create the graph
```

`EvenLessHappyLittleGraphs.py` som først lager ordbøker av datafilene sånn at vi har et oppslagsverk for å hente ut og bearbeide informasjonen litt. Vi bruker ID-som nøkler. For skuespillerne er innholdet en liste med navn og en liste med ID-ene til filmer de har spilt i, mens for filmene er det tittelen og dens rating fulgt av en tom liste vi skal oppdatere til å inneholde alle skuespillere som har spilt i den.
Så har du funksjonen som lager selve grafen og tar ovennevnte ordbøker som parametere. Den deklarerer ordbøker som skal inneholder noder (skuespillere), kanter (filmer) og vekter (tittel, rating). Vektordboka holder også styr på hvor mange kanter det er mellom nodene, altså hvor mange filmer to skuespillere har spilt i sammen. Dette foregår i tre operasjoner:

1. Går gjennom skuespillerordboken – Sjekke om filmen(e) de har spilt i er i filmordboken, og legger de til i nodeordboken med deres ID som nøkkelverdi og filmene de har spilt i som vi har data på som innholdsverdi.
2. Går gjennom nodeordboken – Sjekke filmene hver skuespiller har spilt i og legge skuespillerID-ene til i filmordboken
3. Går gjennom filmene i filmordboka og lager kanter mellom alle skuespillernodene.

Når alt dette er gjort teller den over antall noder og noder og kanter og printer de ut ved å gi oss lengden av mengden med noder (skuespillere) og teller over. Den tar også tiden, var spennende å se.

Six Degrees of IMDB

Oppgave 2:

Ganske enkel direkte implementasjon av BFS som returnerer en ordbok med en node som nøkkelverdi og dens forelder som innholdsverdi, minus startnoden som vi har bestemt er roten i treet vi nå skaper. Denne ordboken bruker vi så til å finne korteste vei mellom den satte startnoden og noden vi ønsker å nå. Vi lager en funksjon som sporer seg fra målnoden bakover i ordboken med en nodes foreldre helt til den har funnet rotnoden, og legger alle nodene på vår vei i en liste som vi snur og returnerer. Den kjøres med kommando

```
$ python sixdegreesofbeiken.py
```

Og gir oss

```
sixdegreesofbeiken.py
Vertices:  124030
Edges:  11460348
It took 20.45 seconds to create the graph

Actor                Movie                           Rating Actor
Donald Glover        Lennon or McCartney             5.3    David Morrissey
David Morrissey      Waterland                       6.6    Jeremy Irons
There are 2 degrees of separation between Donald Glover and Jeremy Irons
It took 0.89 seconds to complete the query

Actor                Movie                           Rating Actor
Scarlett Johansson   Avengers: Infinity War          8.4    Letitia Wright
Letitia Wright       Death on the Nile               6.3    Emma Mackey
There are 2 degrees of separation between Scarlett Johansson and Emma Mackey
It took 0.89 seconds to complete the query

Actor                Movie                           Rating Actor
Carrie Coon          Avengers: Infinity War          8.4    William Hurt
William Hurt         The Countess                    6.2    Julie Delpy
There are 2 degrees of separation between Carrie Coon and Julie Delpy
It took 1.67 seconds to complete the query

Actor                Movie                           Rating Actor
Christian Bale       American Psycho                 7.6    Justin Theroux
Justin Theroux       Star Wars: Episode VIII – The Last Jedi 6.9    Lupita Nyong'o
There are 2 degrees of separation between Christian Bale and Lupita Nyong'o
It took 0.87 seconds to complete the query

Actor                Movie                           Rating Actor
Tuva Novotny         Dear Alice                      5.5    Danny Glover
Danny Glover         LUV                             5.9    Michael K. Williams
There are 2 degrees of separation between Tuva Novotny and Michael K. Williams
It took 0.89 seconds to complete the query
```

Chilleste vei

Oppgave 3:

Ganske enkel direkte implementasjon av BFSxDijkstra, men med en liten hjelpefunksjon for når skuespillere har spilt i flere filmer sammen som finner ut hvilken a de som var best. Hjelpefunksjonen går gjennom filmene gitt to skuespillere og returnerer indeksen til den med høyest rating som vi da henter ut når Dijkstra trenger vekt. Kjøres med kommando:

```
$ python sixdegreesofdisneyplusandthrust.py
```

Og gir oss

```
Vertices:  124030
Edges:  11460348
It took 22.41 seconds to create the graph

Actor               Movie played in              Rating Actor
Donald Glover       The Martian                  8.0    Enzo Cilenti
Enzo Cilenti        The Man Who Knew Infinity    7.2    Jeremy Irons
Weight: 4.8
It took 13.08 seconds to complete the query

Actor               Movie played in              Rating Actor
Scarlett Johansson  Avengers: Infinity War       8.4    Ariana Greenblatt
Ariana Greenblatt   Barbie                       7.1    Emma Mackey
Weight: 4.5
It took 12.80 seconds to complete the query

Actor               Movie played in              Rating Actor
Carrie Coon         His Three Daughters          8.5    Elizabeth Olsen
Elizabeth Olsen     Avengers: Age of Ultron      7.3    Julie Delpy
Weight: 4.2
It took 14.59 seconds to complete the query

Actor               Movie played in              Rating Actor
Christian Bale      The Big Short                7.8    Brad Pitt
Brad Pitt           12 Years a Slave             8.1    Lupita Nyong'o
Weight: 4.1
It took 12.51 seconds to complete the query

Actor               Movie played in              Rating Actor
Tuva Novotny        Borg McEnroe                 6.9    Demetri Goritsas
Demetri Goritsas    Saving Private Ryan          8.6    Paul Giamatti
Paul Giamatti       12 Years a Slave             8.1    Michael K. Williams
Weight: 6.4
It took 23.84 seconds to complete the query
```

Komponenter

Oppgave 4:

Her foretar vi et lett modifisert DFS som tar utgangspunkt i en tilfeldig startnode vi gir til den. I tillegg til å holde styr på hvor den har vært tar den en kopi av mengden noder (skuespillere) som parameter og fjerner fra den alle nodene den har besøkt.
Vi lager så en ordbok vi tenker å fylle med størrelsen på listen over noder vi har besøkt som nøkkelverdi, og hvor ofte de forekommer som innholdsverdi.
Siden alle nodene vi har besøkt nå er fjernet fra mengden noder (skuespillere) kan vi finne en ny tilfeldig node og kjøre vår modifiserte DFS til den er tom. Vi sorterer og stokker litt sånn at vi får en penere utskrift, kjører den via kommandoen

```
$ python components.py
```

og får

```
Vertices:  124030
Edges:  11460348
It took 23.30 seconds to create the graph

There is 1 component of size 120030
There is 1 component of size 15
There is 1 component of size 11
There is 1 component of size 10
There are 3 components of size 9
There are 4 components of size 8
There are 8 components of size 7
There are 12 components of size 6
There are 20 components of size 5
There are 46 components of size 4
There are 124 components of size 3
There are 335 components of size 2
There are 2451 components of size 1

Time to find components: 3.09s
```

Kul oppgave alt i alt.