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. Kjøres med kommando

```
$ python writer.py
Time to write adict:  0.32s
Time to write mdict:  0.8s
graph time:  10327.69
Nodes:  124030
Vertices:  9273516
Time to write:  21.73s
```

writer.py kaller på funksjoner fra kjøtt-og-poteterfilen `HappyLittleGraphs.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.
Til sist har du funksjonen som lager selve grafen og tar ovennevnte ordbøker som parametere. Den deklarerer først en mengde som skal inneholde nodene, en ordbok med mengder som skal holde styr på alle medspillere en skuespiller har hatt, og en ordbok med lister som da inneholder informasjon om disse koblingene – Dvs filmene de har spilt i sammen og dens rating.
Den går gjennom alle nøklene i skuespillerordboka og sjekker om vi har data på minst én av filmene de har spilt i. I så tilfelle blir hen lagt til i mengden noder. Mengder er kjekke her fordi den ikke tar hensyn til duplikater, så vi kjører en sjekk på alle filmer og legger til for alle hen har spilt i.
Mens vi er i løkka sjekker vi alle de neste skuespillerne om de har spilt i noen av de samme filmene, og hvis de har spilt i samme film vi har data på blir det lagt til som en medspiller, og vi henter frem filminformasjonen sånn at vi får vektet og merket kanten.

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, så begynner en liten greie som kommer til å gjøre livet vårt enklere senere: Skriver noder, kanter og vekter til filer vi kan leses med `reader.py` som vil spare oss for en del tid videre i oppgaven.

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

```
Time to read v: 0.07s
Time to read e: 6.25s
Time to read w: 45.60s

Actor                Movie                          Rating Actor
Donald Glover        Lennon or McCartney            5.3    Emma Thompson
Emma Thompson        Beautiful Creatures            6.1    Jeremy Irons
There are 2 degrees of separation between Donald Glover and Jeremy Irons
It took 1.39 seconds to complete the query

Actor                Movie                          Rating Actor
Scarlett Johansson   Asteroid City                  6.6    Margot Robbie
Margot Robbie        Barbie                         7.1    Emma Mackey
There are 2 degrees of separation between Scarlett Johansson and Emma Mackey
It took 1.37 seconds to complete the query

Actor                Movie                          Rating Actor
Carrie Coon          Avengers: Infinity War         8.4    Paul Bettany
Paul Bettany         Avengers: Age of Ultron        7.3    Julie Delpy
There are 2 degrees of separation between Carrie Coon and Julie Delpy
It took 1.29 seconds to complete the query

Actor                Movie                          Rating Actor
Christian Bale       The Prestige                   8.5    Andy Serkis
Andy Serkis          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 1.18 seconds to complete the query

Actor                Movie                          Rating Actor
Tuva Novotny         Eat Pray Love                  5.8    Lidia Biondi
Lidia Biondi         Miracle at St. Anna            6.1    Michael K. Williams
There are 2 degrees of separation between Tuva Novotny and Michael K. Williams
It took 1.18 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

```
Time to read v: 0.07s
Time to read e: 5.97s
Time to read w: 48.74s

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 14.24 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 13.69 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 14.73 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 14.86 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 14.27 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

```
Time to read v: 0.07s
Time to read e: 6.44s
Time to read w: 48.51s

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: 4.13s
```