

Projekt - Järnvägsnät DV1490

Christian Nordahl, Andreas Jonasson, & Joakim Ståhle Nilsson

April 23, 2019

1 Uppgiftsbeskrivning

I detta projekt ska ni arbeta med hantering av nya järnvägsnät i grupper bestående av en eller två personer. Järnvägsnäten kommer att representeras av både riktade och oriktade grafer där noderna är olika stationer och alla länkar har en kostnad av något slag. Ett exempel på detta kan ses i figur 1. Ni ska välja ett av följande tre alternativ.

1. Implementera funktionalitet för att kunna ta fram den billigaste ruten för en resenär att ta sig mellan två specificerade stationer. Vikterna i grafens länkar representerar i detta fallet kostnaden att åka mellan noderna. Lösningen ska fungera för både riktade och oriktade grafer.
2. Implementera funktionalitet för att avgöra det billigaste sättet för att lägga räls samtidigt som att alla de planerade stationerna är sammanlänkade. Vikterna i grafens länkar representerar i detta fallet kostnaden att lägga räls mellan noderna. Lösningen behöver enbart fungera för oriktade grafer.
3. Implementera funktionalitet som kan hitta en rutt gjord för turister sådan att den kör alla spår en och endast en gång om en sådan rutt existerar. Start och slutposition ska avgöras av implementationen. Vikterna är i detta fallet oanvända. Lösningen behöver enbart fungera för oriktade grafer.

För projektet finns det inget givet API som ska följas utan ni får fritt göra er egen design gällande hur ni går tillväga för att lösa problemet, samt hur er lösning implementeras. Ni får använda standardbiblioteken som finns tillgängliga i C++ men inga externa bibliotek eller färdiga lösningar. Er representation av grafen samt lösningen ska vara egenimplementerad men får använda sig av funktionalitet från standardbibliotek internt. Det finns dessutom ett antal medföljande filer som skall användas som indata enligt beskrivningen i nästa sektion. Ert program

ska producera en svarsfil som följer ett visst format. Filformaten finns beskrivna i sektion 1.1.

Förutom ert implementation så ska även en rapport skrivas där ni tar upp bland annat hur ni har löst problemet och vilka algoritmer samt datastrukturer ni har använt er av. För en mer detaljerad beskrivning av rapporten se sektion 2.

Ert program förväntas kunna köras genom kommandotolken och att det tar emot två parametrar. Den första parametern innehåller en sökväg till en giltig indata fil, t.ex. `"C:/Users/JZN/Documents/TrainGraph.txt"`. Den andra parametern är bara relevant om billigaste rutten valet har implementerats och är på formen `"A-B"`, där A representerar utgångspunkten och B målet. Ett exempel på hur exekveringen av ert program kan utföras är

```
.\\StudentSubmission "C:/Users/JZN/Documents/TrainGraph.txt"
"Good Town-Great City"
```

Där StudentSubmission är ert program, första texten inom citattecken är en sökväg till en existerande och giltig fil, och andra texten inom citattecken är anvisningar som enbart är relevanta om billigaste rutten har implementerats.

Observera! Ert program kommer få tre parametrar som input, där första

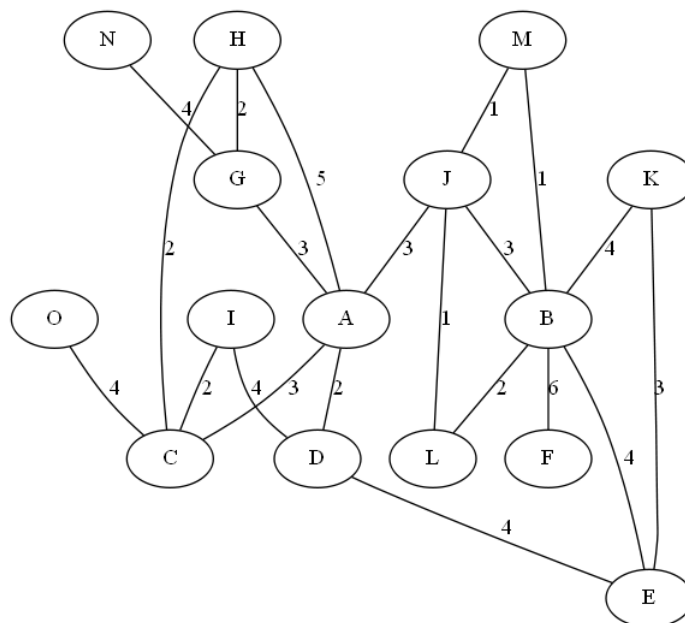


Figure 1: Exempel på en oriktad graf innehållandes noder och länkar med vikter

parametern är sökvägen till ert program. Denna första parameter kan ignoreras.

1.1 Filformat

Ert program ska läsa in den data som finns i filen, bearbeta den, och sedan skriva ut ett svar till en textfil döpt *Answer.txt*. De filer som ska användas som indata är uppbyggda på följande vis

- En rad bestående av antingen “DIRECTED” eller “UNDIRECTED” som indikerar om grafen är riktad eller inte.
- En newline-separerad lista med alla noder som finns i grafen, avslutas med en tom rad
- En newline-separerad lista med alla länkar på formen “A B vikt” där A och B är noder som fanns i den första listan, och vikt är vikten på länken. Om grafen är riktad så går länken från A till B. De tre olika delarna är separerade med tabbar.

För ett exempel på hur en indata fil kan se ut, vänligen se Appendix A.

Svarsfilen ska vara formaterad enligt följande information.

- Om ni implementerat en lösning för problemet gällandes att ta fram billigaste rutt för en resenär mellan två specificerade stationer ska filen se ut enligt följande:
 - Siffran 0 följt av newline
 - Totala kostnaden för rutten följt av newline
 - En följd av stationer separerade med “ -> ” som börjar med stationen A och slutar med stationen B. Följden representerar den för resenären billigaste rutten från station A till station B t.ex. “A -> X -> Y -> Z -> B”
- Om ni implementerat en lösning för problemet gällande att ta fram billigaste sättet att lägga räls ska filen se ut enligt följande:
 - Siffran 1 följt av newline
 - En följd av data på samma sätt som den indata som användes men som nu enbart innehåller de länkar som ska vara kvar.

- Om ni implementerat en lösning för problemet gällandes att ta fram en rutt för turister så ska filen se ut enligt följande:
 - Siffran 2 följt av newline
 - en följd av stationer separerade med “ $->$ ” där följden representerar rutten som ska tas, t.ex. “V $->$ W $->$ X $->$ Y $->$ Z”. Om det inte går att skapa en rutt för den givna grafen så ska “NO PATH FOUND” skrivas istället.

För exempel på hur svarsfilen skall se ut, vänligen se Appendix B.

2 Rapport

Projektet innehåller förutom er implementation även en rapport som ska skrivas. Denna rapport ska skrivas individuellt och görs alltså **inte** tillsammans med någon eventuell gruppmedlem. Din rapport ska vara väl strukturerad och skriven på antingen svenska eller engelska. Ta gärna en titt på följande *länk* för bra information gällande hur ni bör gå tillväga. Det skall tydligt framgå i rapporten om du har arbetat tillsammans med någon och i sådana fall vem du har arbetat med. Rapporten ska innehålla följande delar.

- Beskrivning av problemet
 - Problemet ska brytas ner till delproblem där det är lämpligt
 - Det ska vara tydligt vad det är som behöver göras för att lösa problemet
- Beskrivning och motivering av tillvägagångssätt
 - Algoritmer som använts/implementerats ska beskrivas samt motiveras varför de valdes i jämförelse med andra möjliga alternativ.
 - Datastrukturer som använts/implementerats ska beskrivas samt motiveras varför de valdes i jämförelse med andra möjliga alternativ.
 - Tillvägagångssättet, hur ert program bearbetar den data som matas in genom filen och producerar ett slutresultat, ska beskrivas i sin helhet.
- Analys av implementationen
 - Hur förhåller sig de algoritmer och datastrukturer som du använt dig av i din implementation emot det som förväntas av dessa? Är dessa snabbare/långsammare än förväntat? Förklara varför i så fall samt motivera det.
- Reflektioner över arbetet
 - Hur mycket tid har du lagt ner på projektet och hur är tiden fördelad?
 - Vilka förberedelser gjordes och hur påverkade dessa arbetet?
 - Presterade de algoritmer och/eller datastrukturer som implementerades som du förväntade dig? Om inte, varför gjorde de inte det?
 - Vilka val (algoritmer, datastrukturer, etc) hade du gjort samma/annorlunda om du i nuläget skulle börjat om på projektet?

3 Projektgenomförande

- Det är tillåtet att utföra projektet antingen ensam eller i grupper av två personer
 - Om projektet utförs i grupp så ska båda deltagarna vara aktivt deltagande i projektet.
- All kod skall vara framtagen och implementerad av gruppen.
 - Läs noggrant igenom informationen på sidan **Plagiering** på Canvas.

4 Inlämning och examination

- Kod samt rapport skall lämnas in.
 - Ett **rensat** projekt skall lämnas in tillsammans med utförliga kompilersinstruktioner i .zip-fil format.
 - * Detta betyder att endast källkod och för kompileringen nödvändiga filer skall skickas in.
 - * Inga binära filer eller exekveringsfiler skall skickas med.
 - Detta projekt **skall** gå att kompilera och exekveras från start till slut på skolans datormiljö.
 - Rapporten **skall** lämnas in i .pdf-format namngivet enligt följande:
kurskod_studentakronym_projektrapport
 - * Exempel: DV1490_jonf13_projektrapport
 - * Rapporten skall lämnas in tillsammans med koden men separerat från zip-filen.

A Filuppbyggnad: Indata

UNDIRECTED

Alpha

Beta

Gamma

Delta

Epsilon

Alpha	Epsilon	3
-------	---------	---

Alpha	Beta	2
-------	------	---

Beta	Gamma	3
------	-------	---

Gamma	Delta	5
-------	-------	---

Delta	Epsilon	3
-------	---------	---

Figur 2 visar den graf som bildas av ovanstående fil.

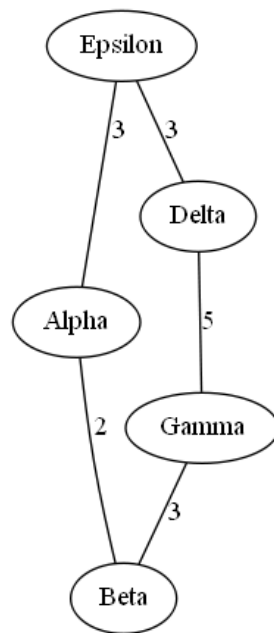


Figure 2: Den graf som bildas av exempelindata filen

B Filuppbyggnad: Utdata

Exempel på svarsfil för billigaste rутten

```
0
8
V1 -> V4 -> V5
```

Exempel på svarsfil för billigaste sättet att lägga räls

```
1
UNDIRECTED
V6
V2
V7
V1
V3
V4
V5

V6 V7 1
V2 V1 2
V7 V4 4
V7 V5 6
V1 V4 1
V3 V4 2
```

Exempel på svarsfil för att ange en rutt som går mellan alla stationer maximalt en gång per länk

```
2
f -> e -> a -> b -> e -> c -> a
```

Exempel på svarsfil för att ange att det inte finns en rutt som går mellan alla stationer maximalt en gång per länk

```
2
NO PATH FOUND
```