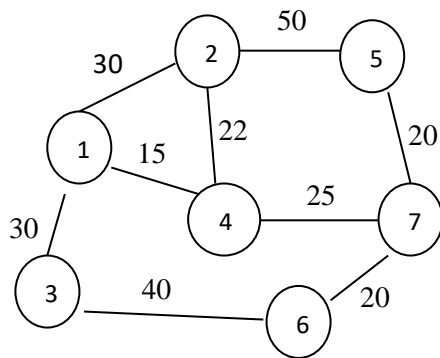


## 811312A Tietorakenteet ja algoritmit 2017-2018, Python-kielinen harjoitustyö

**Tämä harjoitustyö toteutetaan Python-kielillä.** Erään maan teillä on paljon painorajoitettuja siltoja. K. on kuorma-autoyrittäjä maan eräässä kaupungissa ja hän kuljettaa kuormia kotikaupungistaan muualle maahan. K:n ainoa **kuorma-auto painaa 8 tonnia** ja **sen kuorman painolla ei ole ylärajaa**. Ottaessaan työn vastaan K. haluaa tietää, kuinka painavan kuorman hän voi viedä päämäärään kertakuljetuksella. Tätä varten reittivaihtoehdot muotoillaan suuntaamattomana verkkona, jonka solmuja ovat kaupungit ja väleinä niiden väliset tiet. Kaikkien kaupunkien välillä ei kulje suoraa yhteyttä. Väli painotetaan tiellä kuljetettavan maksimipainon mukaan. Painot merkitään täysinä tonneina. Kaupungit numeroidaan kokonaisluvulla alkaen luvusta 1 siten, että 1 on aina kuorman lähtöpaikka. Esimerkiksi alla olevassa 7 kaupungin verkossa



maksimikuorma, jonka voi kuljettaa kaupunkiin 7, on 14 tonnia ja paras reitti on 1->2->4->7. Reitillä voi kuljettaa korkeintaan 22 tonnia ja auto painaa 8 tonnia. Muille reiteille sattuu ankarampi painorajoitus.

Tässä työssä laaditaan autoilija K:lle apuohjelma, jolle syötetään tieverkko rajoituksineen tekstitiedostona seuraavasti: Ensimmäisellä rivillä on välilyönnillä erotettuna kaupunkien lukumäärä ja teiden lukumäärä. Sitten luetellaan kukin omalla rivillään tiet rajoituksineen lukukolmikkoina: lähtökaupunki, maalikaupunki, painorajoitus. Viimeisellä rivillä on kaupunki, johon reittiä haetaan. Esimerkiksi edellä olevan tieverkon esitys määränpäänään kaupunki 7 olisi tiedostona

```
7 9
1 2 30
1 3 30
1 4 15
2 4 22
2 5 50
3 6 40
4 7 25
5 7 20
6 7 20
7
```

Ensimmäisellä rivillä sanotaan siis, että kaupunkia on 7 ja tieosuksia 9. Näin ollen seuraavat 9 riviä antavat verkon välit ja niiden painot. Viimeinen rivi kertoo määränpään, joten verkosta haetaan reittiä kaupungista 1 kaupunkiin 7. Ohjelmalle annetaan syötetiedoston nimi esimerkiksi käyttäjän syötteenä. Ohjelma tulostaa suurimman mahdollisen kuorman tai ilmoituksen, että reittiä ei ole. Tällainen on mahdollista, ellei verkossa ole lainkaan reittiä kotikaupungista annettuun määränpäähän tai jos kaikkien reittien maksimikuorma on alle 8 tonnia, joka on auton paino.

**Ohjelman tulee selviytyä kohtuullisessa ajassa verkoista, joissa on 50 kaupunkia.**

**HUOM! Ongelmaa EI VOI ratkaista käymällä kaikki verkon reitit läpi, koska jo 20 solmun verkossa voi olla yli 6 000 000 000 000 000 erilaista reittiä kahden solmun välillä. Näin ollen ratkaisut, joissa generoidaan kaikki reitit rekursiivisesti, palautetaan automaattisesti korjattavaksi.**

**Opastus:** Ratkaisussa on syytä käyttää jotakin sopivaa tunnettua verkkoalgoritmia. Ongelmaa voi tarkastella esimerkiksi määränpääsolmun saavutettavuusongelmana!

**Arvioi ohjelmasi aikakompleksisuutta**, kun syötteen koon mittana käytetään kaupunkien lukumäärää. Arvioi tämän perusteella, kuinka monen kaupungin muodostamia verkkoja ohjelmalla voidaan käsitellä. Mittaa myös ohjelmasi suoritusajkoja testisyötteillä ja tee tämän perusteella vastaava arvio kaupunkien lukumäärän ylärajasta.

**Ohjelmakoodin lisäksi palautetaan työselostus**, jossa kuvataan

1. ratkaisu ja
2. ohjelman suorituskyvyn analyysi, kuten edellä mainittiin.

Analyysi sisältää siis käytetyn algoritmin analyysin ja ohjelman suoritusajkojen mittaukset.

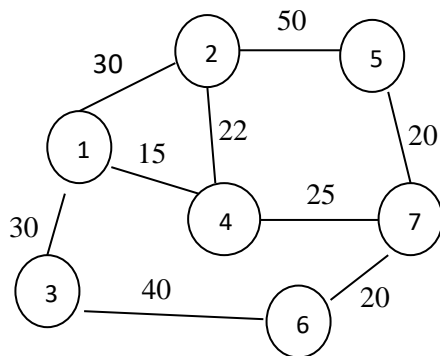
Ohjelmakoodin tulee olla asiallisesti kommentoitu, mutta muuta dokumentointia ei vaadita. Työselostukseen liitetään luonnollisesti myös nimi ja opiskelijanumero. Mikäli haluat, voit antaa työselostuksessa myös palautetta työstä, esimerkiksi tehtävään käytetty työmäärä, työn helpot ja hankalat asiat ja mitä opit työtä laatiessasi. Muutakin palautetta voi antaa.

**Muista pakata tiedostot yhdeksi pakkaukseksi ennen palauttamista.**

**Tehtävän ratkaisu on palautettava viimeistään 15.2.2018.**

## 811312A Data Structures and Algorithms 2017-2018, Python Assignment

**This assignment shall be implemented in Python.** There exists a country with many roads containing weight restricted bridges. K. is a truck owner in a country's city and he transports cargo from his home city to country's other cities. His only **truck weighs 8 tons** and **there is no maximum weight for its cargo**. When he takes a transport task, he wants to know, how many tons of cargo he can take to the destination with one trip. For this, all route options are formulated as an undirected graph, whose nodes are cities and edges are roads connecting them. The weight of an edge will be the maximum weight that can be transported using the corresponding road. There is necessarily no direct road connecting a pair of cities. Weights are tons in integers. The cities are numbered with consecutive integers starting from 1, where 1 is always K's home city and thus the origin of the route. For example, in the following network of 7 cities



the maximum cargo that can be transported to city 7, weighs 14 tons with the optimal route 1->2->4->7. One can transport totally 22 tons and the truck weighs 8 tons. Other routes contain more severe restrictions.

In this assignment you shall help K by implementing a program that takes as an input the road network and its restrictions as a text file as follows: First line of the file will contain two integers: the number of cities and the number of road segments. The next lines will each contain three integers: first two are city numbers and the third is the maximum weight that can be transported on the road between these cities. Last line will contain one integer representing the destination city. For example the previous network with destination city 7, would be input as the following text file:

```
7 9
1 2 30
1 3 30
1 4 15
2 4 22
2 5 50
3 6 40
4 7 25
5 7 20
6 7 20
7
```

First line tells that there are 7 cities and 9 roads. Thus, the following 9 lines define the roads and their maximum weights. The last line tells that we want to know the best route from city 1 to city 7. The name of the input text file can be given as a user input to the program. The program shall print the maximum weight of the cargo or announce that there is no possible route. This latter can happen, if there are no routes from origin to destination or if every route has a maximum weight less than truck's weight (8 tons).

**Your program should be able to handle graphs with up to 50 cities in a reasonable time.**

**NOTE! It IS NOT possible to solve the problem by going through every route between the two cities, because already a graph with 20 nodes can have more than 6 000 000 000 000 000 different routes between two nodes. Hence, all solutions that generate recursively all routes between the cities will automatically be returned for revision.**

**Hint:** Use some well-known graph algorithm. The problem can, for example, be considered as a connectivity problem!

**Estimate the time complexity of your program**, when the size of the input is the number of cities in the road network. Based on this, estimate the maximum number of cities your program could deal with. Also, measure the running times of your program with test inputs and make a corresponding estimate for the number of cities, based on the results of your measurements.

**In addition to program code, you shall return a report** that describes

1. the solution, and
2. analysis of program's performance as mentioned above.

Analysis contains thus the analysis of the used algorithm and the measurements of program's running times. The code shall be commented but no other documentation is required. The report shall naturally contain student's name and student number. If you wish, you can also give feedback in your report.

**Remember to zip the files before returning.**

**The assignment shall be returned by 15.2.2018.**