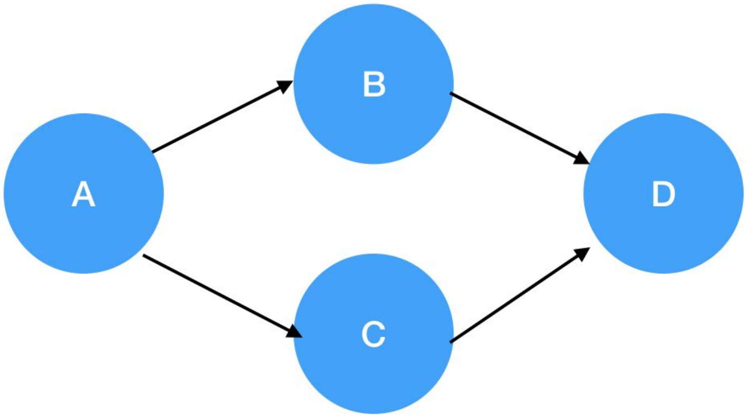
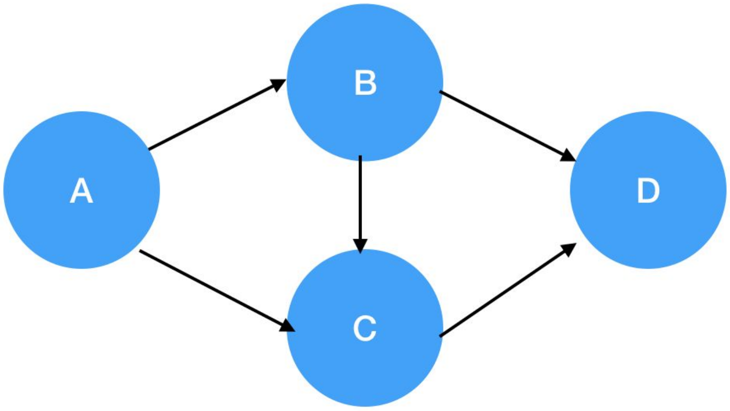
1. Best Routes

At Carvana, vehicles are held in regional inventory locations, and sold on Carvana.com, and then delivered to customers by Carvana trucks. Carvana operates a nationwide logistics network, with trucks running between certain locations of the network on a daily basis. Trucks are running on a fixed schedule, similar to trains. If there is no direct trip available from one location to another, the vehicle will need to be carried and relayed by multiple trips/trucks. Consider a simple network where we have 4 locations A, B, C and D. There are trips running from A to B, A to C, B to D and C to D. If the vehicle is at location A, and the customer is expecting this vehicle at location D. Then we can either send this vehicle via A->B, B->D routes, or alternatively A->C, C->D routes.



Let's say we add one more trip from B->C. Now we have even more alternate options to get the vehicle from A to D.



Exercise:

In the attached data files, *locations.csv* contains a list of nodes (i.e. locations) on the network; *trips.csv* gives you a list of truck trips (or links) which connect the nodes. Please note that all trips listed in the table are one-way. That is, the trip only runs from origin to destination, and the trips are unidirectional.

Please write a class ***RoutesFinder*** in Python to generate the top 3 best routes for any given origin and destination. If there are less than 3 possible routes, export all available options.

Hints:

1. We encourage you to use any existing Python libraries in this exercise.
2. You will need to define a criteria of how to rank possible routes. It can be distance, travel time, or number of stops, etc.
3. For simplicity, 1) if you decide to go with distance, please use vincenty formula to calculate distance between any two points. 2) no need to take traffic into consideration.
4. In your code/output, please use “Origin_Destination” to represent a direct link, e.g. BM_MON represents trip running directly from BM to MON.

Example 1:

Inputs:
Origin: BM
Destination: AUSVM

Outputs:
Option 1: BM_AUSVM
Option 2: NA
Option 3: NA

Example 2:

Origin: BM
Destination: NASH

Outputs:
Option 1: BM_MON, MON_BIR, BIR_WIND, WIND_NASH
Option 2: BM_MON, MON_BIR, BIR_WIND, WIND_LEB, LEB_NASH
Option 3: BM_PERXD,PERXD_KSC,KSC_STLXD,STLXD_INDI,INDI_LEB,LEB_NASH

Please run your script against the following test cases, and submit your outputs along with your code. The output for each given origin-destination pair should be an ordered list of possible routes.

Test case 1 inputs:
Origin: BM
Destination: BOS

Test case 2 inputs:
Origin: WIND
Destination: INDI

2. Operating Hours Calculation

There is a warehouse in CarvanaLand. Trucks come and leave at anytime of the day, and are loaded with packages at the warehouse. Behind the scene, there is an online platform that continuously taking in new orders from customers. Each order comes with a delivery due date. The manager at the warehouse can generate an optimal plan which guarantees all packages arrive at customers on time.

There are also loaders working at the warehouse, who help load trucks with packages. One loader can only load one truck at a time. It takes a loader several work hours to fill a truck. Before a loader starts to load a truck, the manager is expected to give him/her a list of delivery packages that need to be loaded to the truck. Please note that:

1) **Each loader has his/her own work schedule.**

   For example, some loaders work from 8:00 am to 2:00 pm Monday through Saturday, while some loaders work from 10:00 am to 6:00 pm Tuesday to Sunday.

2) **The time it takes to load a truck differs from truck to truck.**

   For example, experienced loaders may be able to load a truck in 3 hours, while other loaders need 4 hours.

3) **Moreover, some loaders prefer loading all packages to a truck on the same day, while some loaders are willing to load a truck across multiple days.**

   For example, a truck leaves at 10:00 am tomorrow, and the loader works from 8:00 am to 4:00 pm everyday, assuming loading the truck takes 3 hours. If the loader prefers doing all the loading work on the same day, then it means the manager needs to give this loader the list of deliveries before 1:00 pm today. So that the loader will have 3 continuous work hours (1:00 pm ~ 4:00 pm) on the previous day to load the truck. But if the loader is willing to divide loading workload into two consecutive days, then he can start loading at 3:00pm today, work for one hour, and continue with the rest of the loading work 8:00 ~ 10:00 am tomorrow. The difference between the two options/configurations is whether the loading time has to be a continuous chunk of time or not.

Exercise:

Write a python script that takes the following inputs:

1. Loader work hours
2. Loader work days
3. Total time the loader needs to load the truck

4. Truck departure datetime
5. Does the loading time have to be on the same day or not?

The script is expected to export a datetime string which tells the manager when she should give the loader a list of delivery packages.

Example 1:
Inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Saturday
3. Loading Time: 3 Hours
4. Truck departure time: 2018-08-13 10:00 (CarvanaLand local timezone)
5. Loading time has to be on the same day

Expected Output: 2018-08-11 11:00

Explanation: The loader has only 2 hours prior to truck departure, and the previous day is a Sunday. So the loader will need to start loading the truck on Saturday afternoon.

Example 2:
Inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Saturday
3. Loading Time: 3 Hours
4. Truck departure time: 2018-08-14 10:00 (CarvanaLand local timezone)
5. Loading time can be spread across different days

Expected Output: 2018-08-13 13:00

Explanation: The loader has 2 hours prior to truck departure, which he/she can use for loading. He/she needs one more hour from the previous work day to complete the loading. The day before truck departure is a regular workday for this loader. So the loader needs to start loading at 13:00 the day before.

Please run your script against the following test cases, and submit your outputs along with your code. All outputs are expected to be datetime strings in “%Y-%m-%d %H:%M” format.

Test case 1 inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Friday

3. Loading Time: 3 Hours
4. Truck departure time: 2018-08-15 14:00
5. Loading time can be spread across different days

Test case 2 inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Friday
3. Loading Time: 3 Hours
4. Truck departure time: 2018-08-15 14:00
5. Loading time cannot be spread across different days

Test case 3 inputs:
1. Work hours: 12:00 ~ 18:00
2. Work days: Monday, Wednesday, Friday
3. Loading Time: 3 Hours
4. Truck departure time: 2018-08-15 10:00
5. Loading time can be spread across different days

Test case 4 inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Saturday
3. Loading Time: 3.5 Hours
4. Truck departure time: 2018-08-14 10:00
5. Loading time can be spread across different days

Test case 5 inputs:
1. Work hours: 08:00 ~ 14:00
2. Work days: Monday through Saturday
3. Loading Time: 3.5 Hours
4. Truck departure time: 2018-08-14 10:00
5. Loading time cannot be spread across different days