# ADSA Assignment 04 – London Tube Trip Planner

The London Underground Network[[1]](#footnote-1), better known as the “Tube”, is the metro system of London, capital of the United Kingdom and one of the most famous cities in the world. Opened in 1863, the Tube is the oldest underground railway system in the world and it has also been the first to deploy electric traction systems train in 1890. It comprises several lines, which are interconnected with a complex system of sub-urban commuter trains.

A map of the Tube is available at: https://goo.gl/images/fNX8yA

Your task in this assignment is to build a simplified trip planner for the Tube using what we have learnt in this course. In particular, you will need to focus on Networkx graphs and the Google Maps API.

In a nutshell, your trip planner should (i) consider the only the “Bakerloo”, “Jubilee” and “Victoria” lines of the Tube, (ii) initialise travelling times between stations using data from Google maps, and (iii) allow users to query the shortest way between two stations, returning also the required travelling time.

Detailed instructions:

**Provided data:**The file “tube\_lines.txt” contains, for each line, the list of stations ordered from end to end.   
(Note that this contains info about all Tube lines, not only the Bakerloo, Jubilee and Victoria!).

Example:

Line: Bakerloo  
Elephant & Castle  
Lambeth North  
Waterloo  
Embankment  
Charing Cross  
Piccadilly Circus  
Oxford Circus  
Regent's Park

...

The file “tube\_locations.txt” contains, for each station, its geographical coordinates (latitude and longitude) and the lines to which it belongs.

Example:

King’s cross is located at latitude 51.53057 and longitude -0.12399 and belongs to many lines:

King's Cross 51.53057 -0.12399 "Piccadilly, Circle, Victoria, Northern, Metropolitan, Hammersmith & City"

Canary Wharf belongs to the Jubilee line only:

Canary Wharf 51.50362 -0.01987 Jubilee

**Step 1 – Building the Tube Graph**

You need to build a graph of the Tube (only Bakerloo, Jubilee, Victoria lines[[2]](#footnote-2)), where each node is a station and edges connect adjacent stations (e.g., see example above, there will be an edge between “Waterloo” and “Lambeth North”, between “Waterloo” and “Embankment” etc.).

Edges should be annotated with travelling times obtained through the Google Maps API (obviously using public transport as travelling mode!).

**Step 2 – Functionality of the Trip Planner**

Once the Tube graph is initialised, your Trip planner should allow the user to:

1. **Plan a trip between two stations (origin and destination)**. This means to:
   1. Calculate the quickest way to get from the origin to the destination and print the list of intermediate stations in the trip
   2. Print the total travelling time.

NOTE: you need not give information about tube “lines”, that is, do not consider that the user may have to change line at some interchange stations and that this may take some time. It is enough to just print a list of intermediate stations between origin and destination.

1. **Update the travelling times to a specific time of the day**. For instance, you can initialise the Tube graph with travelling times captured using departure times at 10am in the morning of a week day, but the user at some point may want to update travelling times using 5pm of a weekend as departure time. Therefore, your trip planner should allow the user to update the Tube graph using specific departure times.  
     
   Alternatively, you may also allow the user to specify a departure time when planning a trip. This looks more similar to what an actual trip planner would allow. In this case, information in the Tube graph will be refreshed live when the user submits a query. Note, however, that refreshing this information using python-gmaps may take some time, so it is acceptable, as mentioned above, to separate the “user query”, i.e., (1), from updating the travelling times, i.e., (2).  
     
   (Beware of time differences between Korea and London! Note also that the Tube is closed at nights, so it does not make much sense to use departure times between 11.30pm and 6.30am, since Google maps will assume you will walk or take night buses!).

Instructions:

* You have to conduct this project in **groups of 2 or 3 people, groups must be the same as Assignment 2**.   
  You have to choose one member of the group to become “the captain”: the captain is the one who will have to make the submission on blackboard.
* You have to **submit the code** that you develop **on blackboard** by the **deadline of Sunday Tuesday December 13th at 10pm**.   
  Please zip the folder “assignment04” on your computer and upload the zip file.  
  The submission area will be available in due time.
* As you usual, you will have to give a demo of your code (in the period Dec 14th-16th). More details will be provided in due time.  
  You are not allowed to change your code between the submission deadline and the demo (we will check!).   
  “Giving a demo” means to show the functionality of your software. So, you have to develop appropriate code in the main() of your application to showcase the implemented functionality. Failing to demonstrates the implementation of (some of the) functionality will lead to point deductions.  
  All group members must be present at the demo and must demonstrate that they know the code.

1. More info: https://en.wikipedia.org/wiki/London\_Underground [↑](#footnote-ref-1)
2. Considering only these 3 lines allows to keep the graph simple. Any cycle in the graph obtained will have non-decreasing travelling time, obviously. [↑](#footnote-ref-2)