Problem Statement

Imagine you work for a small software company that has won the tender to deliver a fuel management software solution to a large Airline company. The airline is entering the European air freight cargo business and wants a software tool to manage their fuel purchasing strategy.
Each week, an aircraft will fly trips to a number of other cities. The airplane will start and end in the same home airport each week. The company has a number of different types of aircraft that have different fuel capacities and the cost of fuel varies from airport to airport. Given a list of airports (including to the home airport) that a given plane needs to visit in a week, the most economic route must be found to visit all the airport and return to the home base.

Specifications

- The distance between airports is calculated as the great circle distance between them
- The cost of fuel is assumed to be 1 euro per 1 litre at Airports where the currency is Euros
- The cost in of fuel in airports where the local currency is not euros is assumed to be the exchange rate from the local currency to euro. e.g. in you travel from London to Dublin and the exchange rate is GBP£1 = €1.4029 and you purchase 1000 litres of fuel it will cost €1402.
- The fuel capacity of all aircraft are given in litres with an assumed fuel capacity of 1 litre per km flown

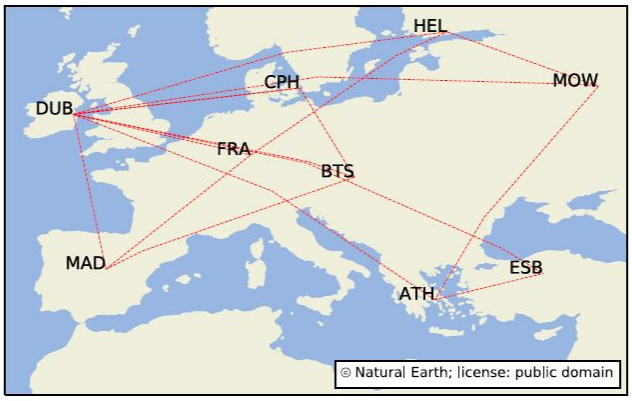
Example

Flight Plan

DUB → LHR → CPH → HEL → MOW → DUB (Boeing 777)

Cheapest Route

DUB → LHR → CPH → HEL → MOW → DUB (€2,883.51)



Example

Computing the Cheapest Route

DUB → LHR → CPH → HEL → MOW → DUB (€2,883.51)

Need to:

1. Check the distance of each leg and ensure the aircraft can make it (a Fokker 50 will not fly from Dublin to Sydney) – Report an error if the route cannot be solved
2. Compute the cost per leg (assume aircraft only fuelled to the capacity required for the next single leg)
3. sum the cost of the full itinerary

Example

AirportCode	DUB	LHR	CPH	HEL	MOW	DUB
CurrencyCode	EUR	GBP	DKK	EUR	RUB	EUR
ExchangeRate	1	1.4029	0.134	1	0.01524	1
Distance	449	979	892	898	2792	
Cost (rate*dist)	449	1373.4	119.5	898	42.5	
Total	€2883					

Hash Tables

Airports

	AirportID	Name	City	Country	IATA	ICAO	Latitude	Longitude	Altitude	Timezone	DST	Tz	Type	Source
417	421	Helsinki Vantaa Airport	Helsinki	Finland	HEL	EFHK	60.317200	24.963301	179	2	E	Europe/Helsinki	airport	OurAirports
502	507	London Heathrow Airport	London	United Kingdom	LHR	EGLL	51.470600	-0.461941	83	0	E	Europe/London	airport	OurAirports
592	599	Dublin Airport	Dublin	Ireland	DUB	EIDW	53.421299	-6.270070	242	0	E	Europe/Dublin	airport	OurAirports
601	609	Copenhagen Kastrup Airport	Copenhagen	Denmark	CPH	EKCH	55.617901	12.656000	17	1	E	Europe/Copenhagen	airport	OurAirports

Country Currency

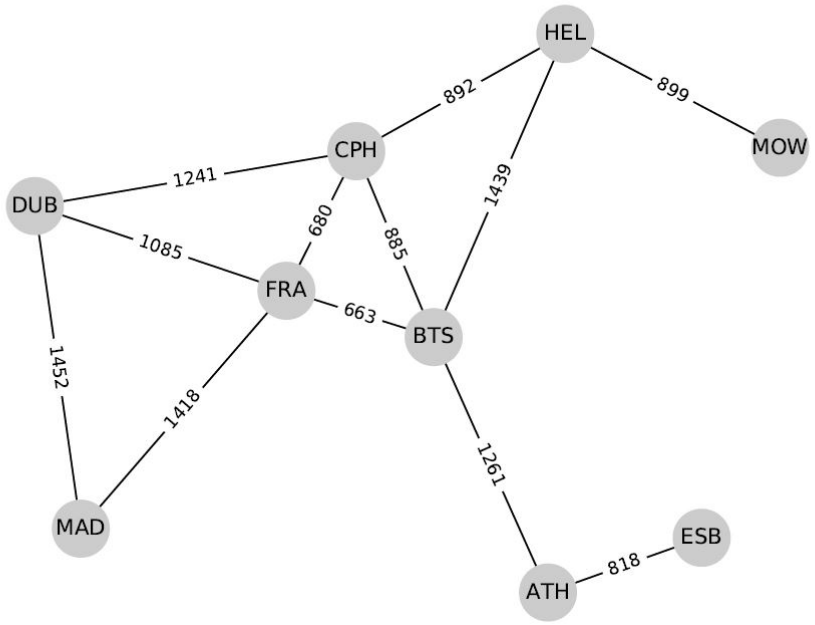
	name	ISO3166-1-numeric	ITU	WMO	Dial	FIFA	currency_alphabetic_code	currency_country_name	currency_name
59	Denmark	208	DNK	DN	45	DEN	DKK	DENMARK	Danish Krone
73	Finland	246	FIN	FI	358	FIN	EUR	FINLAND	Euro
105	Ireland	372	IRL	IE	353	IRL	EUR	IRELAND	Euro
180	Russia	643	RUS	RS	7	RUS	RUB	RUSSIAN FEDERATION	Russian Ruble
233	United Kingdom	826	G	UK	44	ENG,NIR,SCO,WAL	GBP	UNITED KINGDOM	Pound Sterling

Currency Rates

	Currency	CurrencyCode	toEuro	fromEuro
45	Danish Krone	DKK	0.13400	7.4614
53	Euro	EUR	1.00000	1.0000
58	British Pound	GBP	1.40290	0.7130
139	Russian Rouble	RUB	0.01524	65.6681

Graphs

Nodes, Edges and Weighted Edges



Term Project: Functional Components

Interfaces

File I/O:Reading and Writing From a csv, User interface: Text and File based

Data Structures: Itinerary management

Creating and managing the route data

Data Structures: Lookups and Data Management

Storing lookup date and accessing conveniently, efficiently (AirportAtlas, Currency etc.)

Algorithms: Pricing Calculator

Algorithms to calculate the best price for a given route

Data Structures and Algorithms

- Object-oriented programming classes to model the
- high level data structures
- Appropriate use of Python objects

Target 2 Data Structures/Algorithms to implement/ optimise

Example Classes and Data Structures

Airport, Aircraft, Airport Atlas (for looking up airport details), Flight Plan
Dictionary/Hash Table/Graph

Examples of Possible Algorithms

Routing Algorithm, Pricing Algorithm Dijstrika method/ floyd-warshall method

Developing your solution

Design and Development strategy

Unit test methods and error checking/exception handling.
Evaluate Algo/Data Structures vs Generic Python alternative.
Make design decisions as if this was a real system, e.g. consider assumptions like exchange rates may need to be updated daily.

Deploy in modules for integration
Caching
Search/Insert/Delete trade offs

When you deliver the project, it should have these 4 parts.

1.Object-oriented programming (OOP) should be used to model the high-level data and solve the problem.

2.A physical data structure refers to the actual organization of data on a storage device.

Physical Data Structures (: Array/Linked list)

3.The logical data structure refers to how the information appears to a program or user

Logical Data Structures: Stack/Queues/Trees/Dictionary/Hash Table/Graph

---

4.Algorithms may be useful for this project: Routing Algorithm, Pricing Algorithm, Dijkstra Algorithm .

The project should be solved in 2 Data Structures/Algorithms combinations.
For example, if there are 3(you can design any combination as you like,as long as it works) combinations suitable, solve it in every 2 of them below will be ok:

OOP /Array/ Stack/ Routing Algorithm
OOP / Linked list / Stack/ Routing Algorithm
OOP /Linked list/ Queues / Dijkstra Algorithm

Do not worry about these 4 parts/combination, I just give you guidance. If you do the job , these 4 parts are just 4 procedures that you will automatically go through.