**Practice Enterprise 2**

# Route Planning / Green Routing :

## Overview

* Route plannen van cargo tot thuis van klant
* Groene routes vergelijken

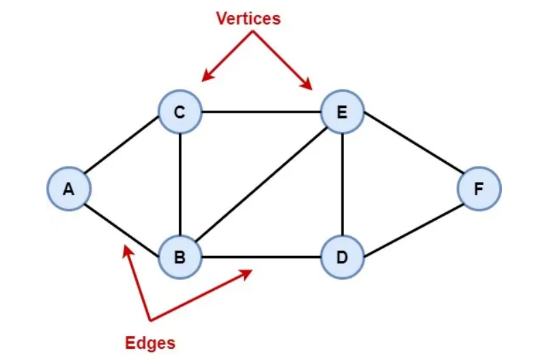
## Find out how to start coding

* Base of a code found in : <https://www.101computing.net/air-flight-route-planner/>
* Basics of Graph data structure in : <https://python.plainenglish.io/graph-data-structure-theory-and-python-implementation-ee8c9795eae7>
* Base of Dijkstra’s shortest path algorithm in : <https://www.bogotobogo.com/python/python_Dijkstras_Shortest_Path_Algorithm.php>
* Greenest route planning in : <file:///C:/Users/vegam/Downloads/Green_Paths_software_metapaper_pre-print.pdf>
* basic python dijckstra <https://pythonalgos.com/dijkstras-algorithm-in-5-steps-with-python/>
* Code written based on: <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

## Extra’s

* Prioriteiten geven voor snel vertrekkende vliegtuigen en grote hoeveelheden pakketjes
* Limiteren van wegen
* Limiteren van totale tijd
* Hoge prioriteit

## Graph data structure

* Non-linear data structure
* Nodes and edges
* Powerful modeling tool to solve real-world problems
* Nodes referred to as vertices
* Formal representation of graph: G = (V (set of vertices), E(set of edges))
* Graph can have following form (undirected):
* In an undirected graph, movement can go to both sides
* A picture containing clock

  Description automatically generatedDirected graph is as follows:
* In a directed graph, movement can go to only one side
* Most cases graphs have a weight value
* Path : sequence of edges that allows us to go from one node to another
* Length of path is defined as the number of edges it contains
* Adjacent matrix or adjacent list

### Adjacent matrix

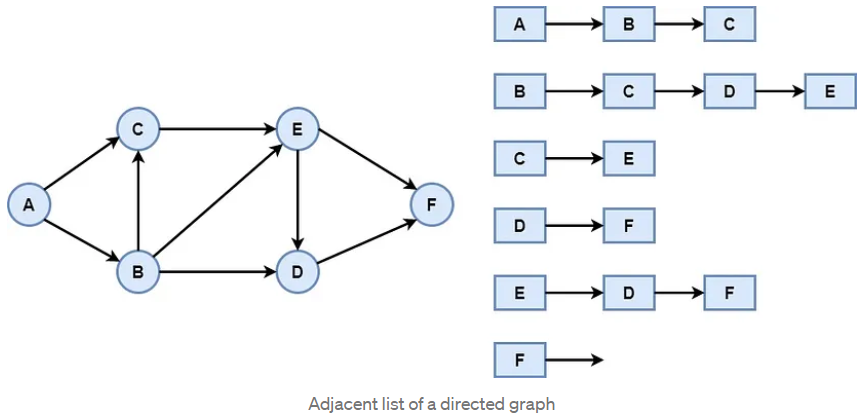
* 2d array used
* Matrix full of values
* Value is etiher 0 or 1
* Edge between nodes is 1, no edge is 0
* Diagram, schematic

  Description automatically generatedExample:
* Diagram

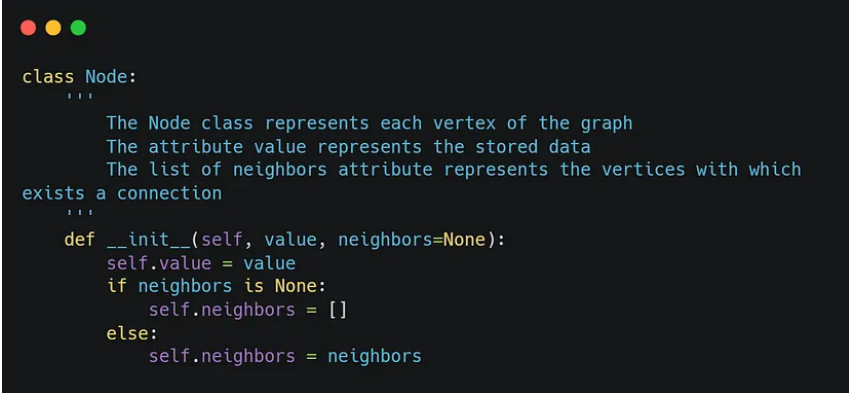
  Description automatically generatedDirected graphs look different :

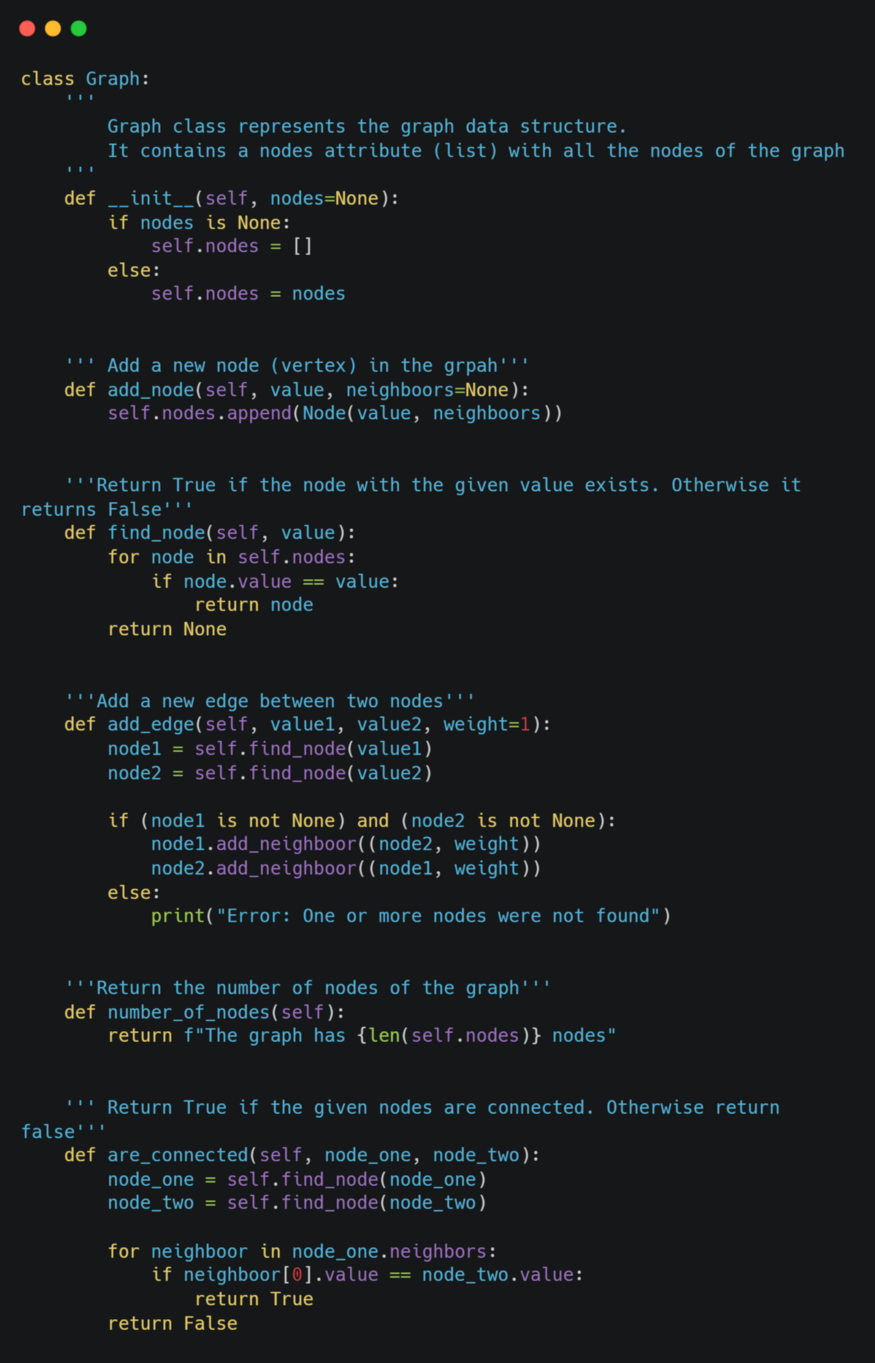
### Adjacent list

* Array of linked lists
* Each vertex of a graph has an index
* This represents the head of the list with all connected nodes
* A picture containing electronics

  Description automatically generatedUndirected example :
* Directed example :

### Python implementation

* Use of classes :
* Text

  Description automatically generatedChecking if it has neighbours, how many and even create new ones :
* Class to represent graph data structure, allows to add nodes and edges between them :

## Dijkstra’s shortest path algorithm

* Two variants : original found shortest path between nodes, common variant fixes source node and finds shortest path from source to all other nodes -> shortest-path tree
* Uses labels that are real numbers or positive integers which are ordered

## Distance between nodes

* Pgeocode library??
* Pgeocode allows you to find the distance between postcodes and provides geo-info

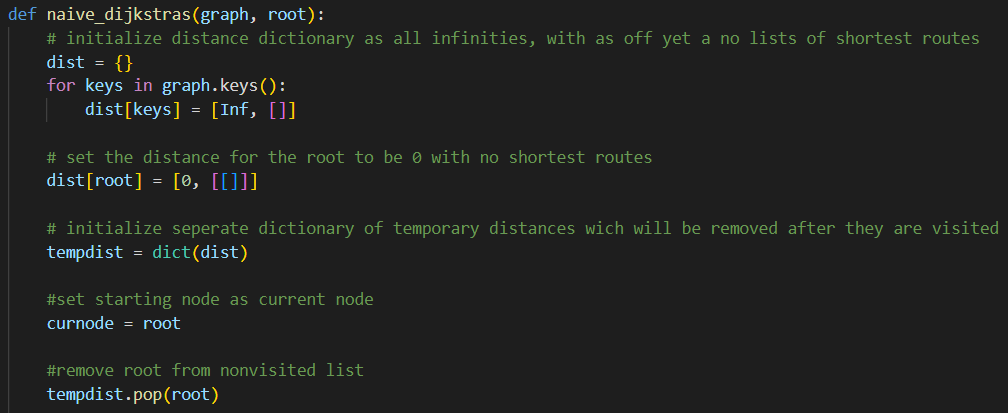
## Code

The graph is a nested dictionary with the first key being the starting node, the embedded key is the destination node and the embedded value is the time it takes to reach the destination. (This may be subject to change)

Afbeelding met tekst, binnen, zwart

Automatisch gegenereerde beschrijving

For the naïve Dijkstra function a root node is required as well as the graph used. It will calculate the shortest possible routes for every node and will return the shortest distance and every possible route via the shortest distance.



Afbeelding met tekst

Automatisch gegenereerde beschrijving

### API

* Google or drivebestway.com?
* Google provides travel distance and time for a matrix of origins and destinations (not free)
* <https://developers.google.com/maps/documentation/directions/get-directions>
* <https://rapidapi.com/collection/directions> free direction api’s
* <https://rapidapi.com/trueway/api/trueway-directions2/>
  + Popularity 9.2/10 ; service level 100%
  + Possibility to put different destinations to calculate the route we need to take
  + Might be only working with coordinates
  + Httpclient, unirest, requests
* <https://rapidapi.com/trueway/api/trueway-matrix/>
  + Popularity 9.6/10 ; service level 100%
  + Also possibility to put various stops, which are the different city’s or villages we go to
  + Also working with coordinates, need to make more research
  + Httpclient, unirest, requests

### Imports

* Numpy
* Pandas
* pgeocode