**Practice Enterprise 2**

# Route Planning / Green Routing :

## Overview

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

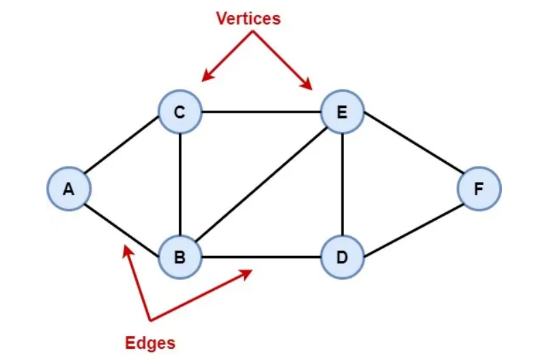
## Find out how to start coding

* 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/>

## 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

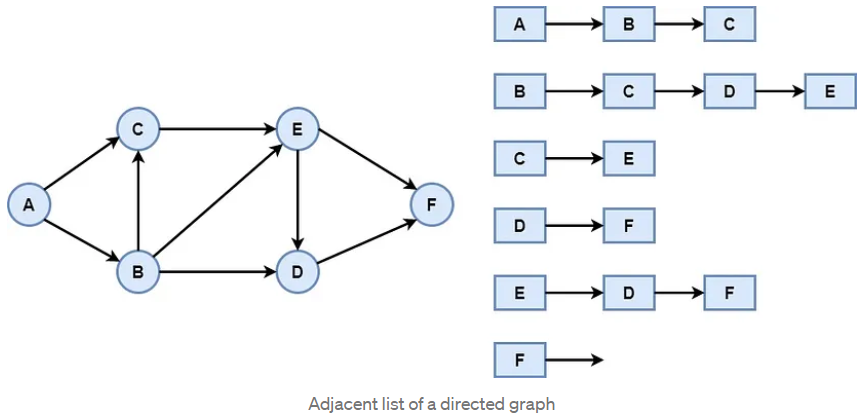
* 2d array used
* Matrix full of values
* Value is etiher 0 or 1
* Edge between nodes is 1, no edge is 0
* Diagram, schematic

  Description automatically generatedExample:
* Diagram

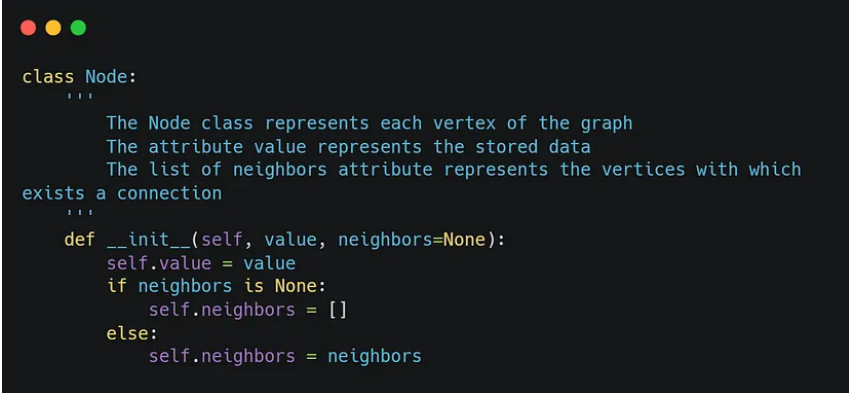
  Description automatically generatedDirected graphs look different :

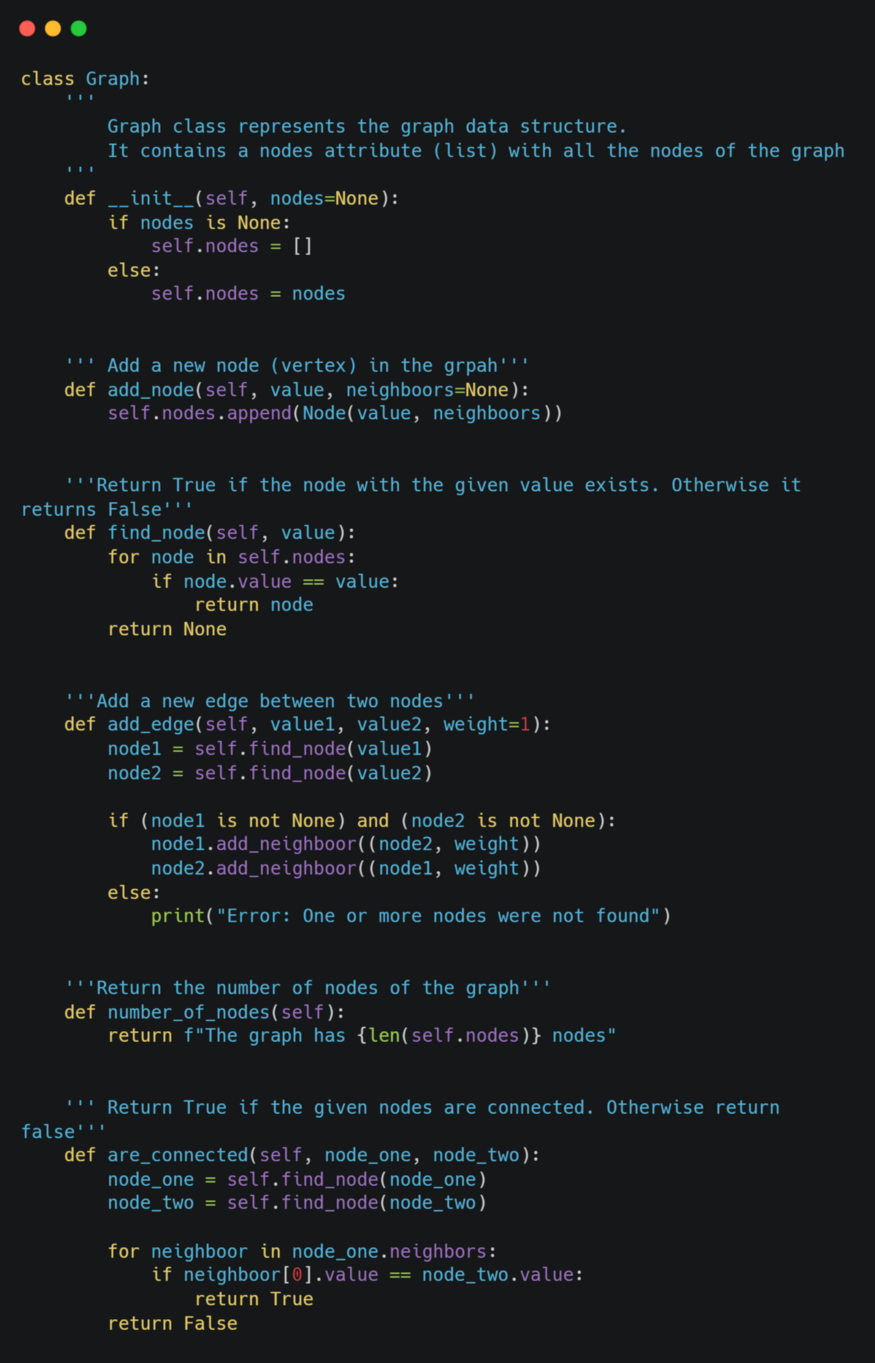
### Adjacent list

* 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

## Distance between nodes

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