Programming fundamentals with Python

Pepe García

2020-04-20

Programming fundamentals with Python

Plan for today

Learn what graphs are

What kinds of graphs exist

Learn how to model graphs with Python

Understand simple graph operations

Graphs

Graphs are data structures to represent relations between objects

Graphs are composed by nodes and edges. Nodes are each one of the objects in the graphs, and edges are the connections

Different kinds of Graphs

There are some features that we should identify about our graph:

Is it **directed**?

Is it **weighted**?

Different kinds of Graphs

What kind of graphs does the following form?

- Twitter follows
- Facebook friendships
- Linkedin contacts
- Metro network

Graphs

Where can we use graphs?

Graphs

Dependencies

Graphs

Königsberg bridges problem

Representing Graphs

There are two main ways of representing graphs in a computer.

- using an **Adjacency matrix**
- using **Adjacency lists**

Adjacency matrices

With this technique we will use an adjacency matrix representing connections between nodes. Ones represent that there's a connection, zeros that there isn't.

	a	b	**c**	**d**
a	0	1	0	0
b	1	0	1	1
c	0	1	0	0
c	0	1	0	0

Adjacency matrices

Can you think of a problem that Adjacency matrices have?

Adjacency lists

Adjacency lists, on the other hand, store a list of all the connections to each node.

```
{
  "A": ["B"],
  "B": ["A", "C", "D"],
  "C": [],
  "D": []
}
```

Graphs

Representing graphs in Python

get all the nodes

get all the edges

Exercises

- Create a function **not_connected** to find non-connected nodes in a graph
- Create a function **fully_connected** that returns True if a graph is fully connected, False otherwise
- Design a way of implementing weighted graphs

Bibliography

https://www.python.org/doc/essays/graphs/

Investigate the networkx library if you've time