Bioinformatics

Discrete Mathematics and Optimisation

Problem Sheet Graphs and Networks (Programming)

---

1. Create a class in `python` to represent graphs. This class should use a dictionary to store all information about the vertices and edges. Furthermore, the class should contain methods for:

   (a) returning lists of the vertex and of the edge set,

   (b) adding vertices or edges.

2. Write a `python` script which, given a sequence S of length $n-2$ with entries in $\{1, 2, \ldots, n\}$, outputs the edges of the tree T that has Prüfer code S.

3. Create a `python` script for the BFS algorithm. Use it to write scripts for the following.

   (a) Given two vertices x, y in a graph G, display a shortest path joining them.

   (b) Given a graph G, count the number of its connected components.

   (c) Given a graph G, decide if G is a tree.

4. Write a `python` script which implements the algorithm described in Exercise 5 of the problem sheet Graphs and Networks II. If the weighting satisfies the triangle inequality, the output should have weight at most twice the optimum.

   Hint: A good way to test your function is to realize that any weighting that is a *metric* will satisfy the triangle inequality. The Wikipedia article [1] has a list of examples.

References

[1] https://en.wikipedia.org/wiki/Metric_(mathematics)