Fundamental Algorithms

Exercise 1 – Hypergraphs

A hypergraph extends the concept of a graph in the sense that edges are allowed to connect an arbitrary number of vertices (instead of exactly two). Hence, a hypergraph is defined as a tuple \((V, H)\), where \(V\) is a set of vertices and \(H\) is a set of hyperedges, where \(H \subset \mathcal{P}(H) \setminus \{\emptyset\}\), with \(\mathcal{P}(H)\) the power set (i.e., the set of all possible subsets) of \(H\).

Let’s assume a hypergraph where \(V\) is a set of authors, and each hyperedge \(h \in H\) contains all authors of a specific scientific article.

a) Give a suitable definition of the concept of a path in a hyperedge.

b) Given is the hypergraph \(S = (V_S, H_S)\) of “all” scientific articles. The Erdős number \(\text{Er}(a)\) of an author \(a \in V_S\) is defined as the length of the shortest path in \(S\) that connects the specific vertex \(e \in V\) (\(e\) corresponds to the author Paul Erdős) to \(a\). Write down an algorithm to determine \(\text{Er}(a)\).

Hint: you can build such an algorithm by extending one of the graph traversals we discussed in the lecture!

c) Try to formulate the problem of 1b) as a graph problem!

Exercise 2 – Bipartite Graphs


The following exercise is based on the concept of so-called bipartite graphs:

A graph \((V, E)\) is called bipartite, if there exist \(V_0\) and \(V_1\) with \(V_0 \subset V\) and \(V_1 = V \setminus V_0\), and for all \((v, w) \in E\) there is either \(v \in V_0\) and \(w \in V_1\), or \(w \in V_0\) and \(v \in V_1\).

To put it simpler: for a bipartite graph, it is possible to attribute each node \(v \in V\) with one of two “colors”, say red and black, such that any edge \(e \in E\) will connect a red and a black node (and no edge will connect edges of the same color).

a) Give a prove to the following claim:

If a graph \((V, E)\) is bipartite, then it cannot contain an odd cycle (i.e., a cycle of odd length).
b) Try to find an algorithm that tests whether a given graph is bipartite.

*Hint: you can build such an algorithm by extending one of the graph traversals we discussed in the lecture!*

c) Try to give a prove for the following claim (using the algorithm from Exercise 2):

*If a graph \((V, E)\) is not bipartite, then it will contain an odd cycle.*

d) Consider the graph in Fig. 1 obtained from a Cartesian discretization mesh. Is this a bipartite graph?