Algorithms Example Sheet 4: Further Reading

Further reading for DFS

There are some core properties of DFS that are not covered in the lecture notes. However, these are essential for some of the more advanced applications of DFS and they are quite intuitive to understand.

Exercise 4.C.1 [pre/post numbers] (optional) Read section 3.2/3.3 from "Algorithms" by S. Dasgupta et al. and answer the questions below:

(a) Define the pre/post numbers for each node in a DFS.

(b) Prove that for any two nodes u and v, $[\text{pre}(u), \text{post}(u)]$ and $[\text{pre}(v), \text{post}(v)]$ are either disjoint or one contains the other.

(c) What can you say if for two vertices u and v, $[\text{pre}(u) \leq \text{pre}(v) \leq \text{post}(v) \leq \text{post}(u)]$?

(d) Prove that a directed graph has a cycle if and only if its depth-first search has a back edge.

Project 1 [Strongly connected components] Read section 3.4 from "Algorithms" by S. Dasgupta et al. (or Section 22.5 in CLRS) and answer the questions below:

(a) Define the *strongly connected components* for a directed graph.

(b) Show that the strongly connected components of a graph for DAG.

(c) Describe how to efficiently compute SCCs in $\mathcal{O}(V + E)$ time.

(d) How can you use SCCs in the following problems:

i. Re-attempt Exercise **??** aiming for an $\mathcal{O}(V + E)$ algorithm.

ii. Given a directed graph G determine if there is any vertex v from which all other vertices are reachable.

(e) (optional) There are two well-known algorithms for finding strongly connected components in linear time: Kosaraju's and Tarjan's algorithm. Read about the algorithm you did not use in your answer for part **??**.

Project 2 [Articulation points and bridges] Given an undirected graph $G = (V, E)$, an *articulation point* is a vertex that when removed will make the graph disjoint. A *bridge* is an edge that when removed will make the graph disjoint.
Attempt either Exercise 3.31(e)-(h) from "Algorithms" by S. Dasgupta et al. or Problem 22.2 in CLRS, to learn about linear time algorithms for identifying these.

Further reading: If you are interested in more theoretical properties of the DFS, then you can also read this chapter.

Shortest paths

Project 3 [Gabow's scaling algorithm] (++) Attempt problem 24.4 to learn more about Gabow's algorithm.