

Computational Foundations II (Summer Term 2022) Tutorial 5

Tasks marked with a star like **Optional Task*** are optional. Tasks marked like **Hard Task⁺** are given, but it is not expected that you solve them now. It is great if you learn to solve them during the lecture. Go back to them after a few weeks and see your own progress.

Learning Outcome: Shortest Paths

Task 11: Dijkstra

Run the Dijkstra Algorithm on Paper using the example from Slide 35

Task 12: C++-Dijkstra and Dot

Compile (use linux, might be easier, Docker or VirtualBox can help if you are on Windows or Mac). Compare your solution to the output of this program.

Task 13: A Baseline Demo

The GISCUP 2015 implementation at <https://github.com/mwernerds/giscup2015> contains a few additional shortest path algorithms. Please ignore them, they are not obvious. However, the whole framework is carefully organized, try to understand it from a software technology point of view.

There is a benchmark part which has no GUI and no useless debugging code to measure speed.

There is a GUI which provides visualization, scrolling, and interaction,

There is a demo mode in the GUI, where random paths are computed.

All of this is based on Boost and wxWidgets, which are complicated, but powerful libraries as they can bridge Linux, Windows and Mac with ease.

Try compiling and see if you succeed. Let us know your challenges. Advanced students should try to cross-compile on Linux for Windows x64 using the MXE M Cross Environment.

OpenGL (plain old OpenGL) is used to visualize things. This can be handy in your future work...