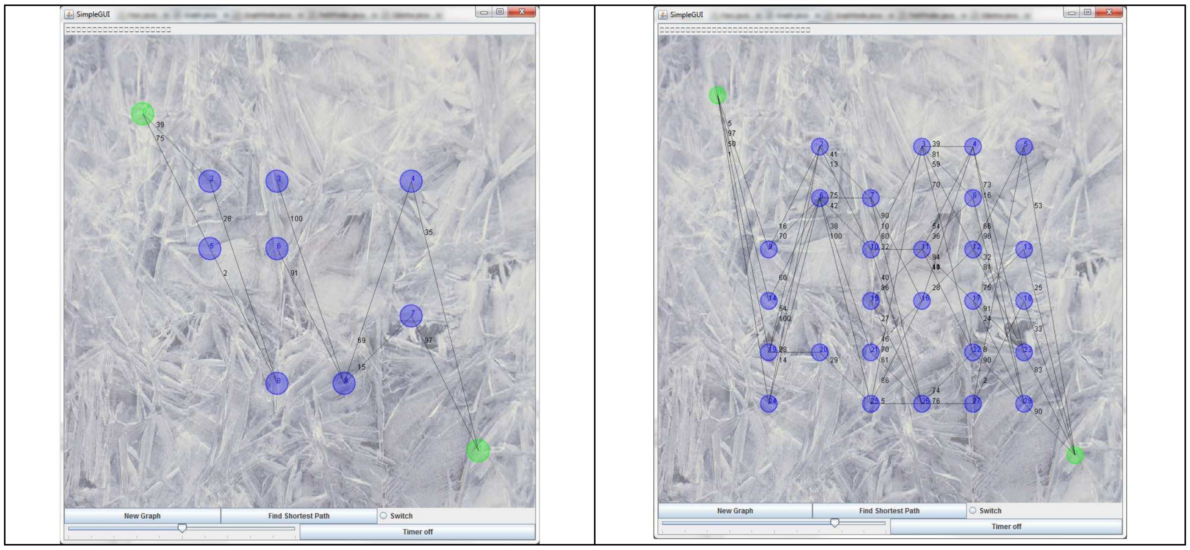
Dijkstra's Algorithm

It's getting serious. You will have to program Dijkstra's algorithm to find the shortest path in a graph. I do not expect you to program an optimized version with heap-implementation etc. It just has to work.
In order to create and visualize a graph, and the path, you will be given a nice interface:



This interface, based on our favorite package, SimpleGUI, creates a path with positive weighted edges on the click of a button. The slider determines the number of nodes. Start and end node for your path algorithm are marked in green.

The graph visualization is part of the project "VisualPath", which already sets up everything for you. You only have to implement the path finding algorithm. In the VisualPath project, you will find two packages, "graph" and "pathfinders". You need to implement your code in the class pathfinders.Dijkstra, in the method "findPath". This method is connected to button 2 of the GUI.

Parameters of method "int[] findPath(Graph g)":
input: Graph g --- a Graph class. You have access to the methods:

- int g.getNumberOfNodes()   // returns number of nodes in Graph g
- int [][] g.getNeighborsOfNode(int nodeIndex) // returns a 2D array of integers, which are the neighbors of the node <nodeIndex>, and the weight of the edge to this neighbor.

output: an integer array, containing a sequence of node-indices. If that sequence is a valid sequence, it will be displayed automatically by the GUI.
The only representation of nodes you get from the Graph class is by index. You need to build your own structure around this. You are not allowed to alter anything in the

package "Graph". I just added the source code to the project to let you look into it, if you are interested (the Graph package is a bit of a hack, I admit).

Your task in detail:

- download the package
- implement the method "findPath"

That's it!

good luck.