Jarnik-Prim Algorithm

* Graph already exists : Nodes, Edges, Costs (implemented with new constructor and print function)
* Find the MST by starting of somewhere at first (Node 0)
* When no edge exists it’s cost is 0 (so minimum edge has to be > 0 when we compare)
* Closed set: visited nodes (Node 0 is in it right away)
* Open set: nodes reachable from nodes in closed set (edge cost between nodes != 0)
* Find the smallest cost to node in open set
* Do this while we haven’t visited all nodes (for example while length(closed set) != number of vertices (Nodes))
* Add all the distances together when we have (20 Nodes) => 19 edges
* Beware of the case when MST doesn’t exist : no more nodes to connect