**Assignment 5**

Minimum Spanning Trees

|  |  |
| --- | --- |
| **Supervisor(s):** | **Roger Munck-Fairwood** |

|  |  |  |
| --- | --- | --- |
| **Student name** | **Student number** | **Signature** |
| James Testmann | S071954 |  |
| Aimo Suikkanen | S082577 |  |
| Lukas Janocko | S135272 |  |
| Shicheng Dai | S133342 |  |
| Sudhir Kumar Chaurasiya | S137239 |  |

* Describe an algorithm (pseudocode) that will give the cheapest routing for connection of the houses in Figure 1 (there must be no cycles).  
    
  prims**(** vertex v **)**

**{** mark **v** as visited and

include in spanning tree**;**

**while(** there are unvisited vertices **)**

**{** find the least**-**cost edge**(**v**,** u**)**

from some visited vertex **v**

to some unvisited vertex **u;**

mark**(** u **);**

add u and edge**(**v**,** u**)** to spanning tree**;**

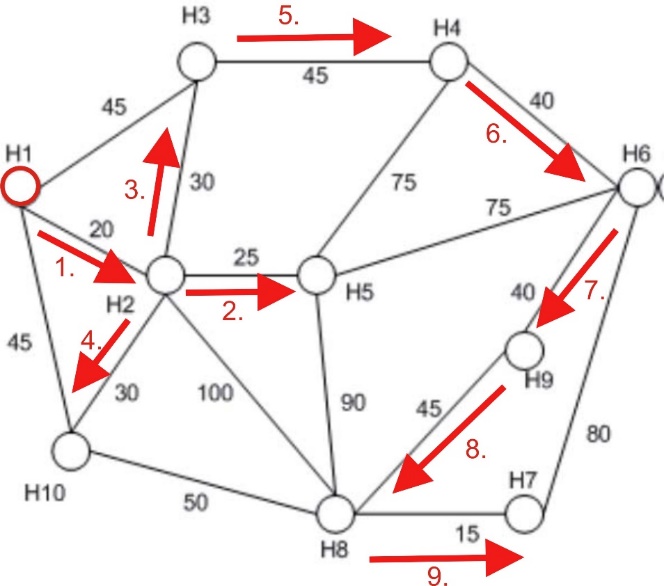
**}** // end while

**}**   
  
From starting point (vertex) we look around for the cheapest edge. We move to the new vertex and we add edge together with vertex to the spanning tree. Then we continue from current vertex, we search for cheapest edge, but we also consider if there is cheaper edges from already visited vertices. We continue until there is no unvisited vertex.

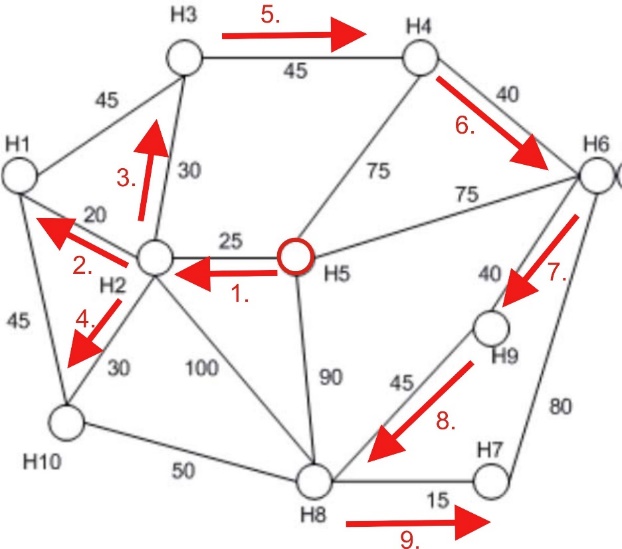
* At least two different houses should be used as starting point (use H1 and H5 to make it easier to compare solutions)

This solution is shown in the next point.

* Run the algorithm by hand, describing the routing point by point together with the total cost (use the following tie‐breaker:  Select the node having the lexographically lowest label).
  + Starting point at H1.





* + Starting point in H5.
* More than one solution may exist. In this case how many solutions can you find?  
    
  There is only one solution. No matter from where we start, the solution will still be the same. Which is the idea of a minimum spanning tree.
* Compare the minimum spanning tree algorithm with the Dijkstra shortest path algorithm   
  and reflect on the principal similarities and differences.

In both Dijkstra and Minimum Spanning Tree, the primary subject is to notice the shortest edge to the connected vertices.

**Dijkstra shortest path algorithm** is preferable when we want to know the cheapest route from starting node to specific one.

**Minimum spanning tree algorithm** is convenient when we want to know the cheapest route through all nodes.

* A graph may be implemented as an adjacency matrix, or an adjacency list. What is the complexity (in big O notation) of the minimum spanning tree algorithm with the two different graph models?

All edges connected to the notes is considered, therefore more edges in a system become more calculations. Therefore