Assignment 5

Graphs: 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**;**

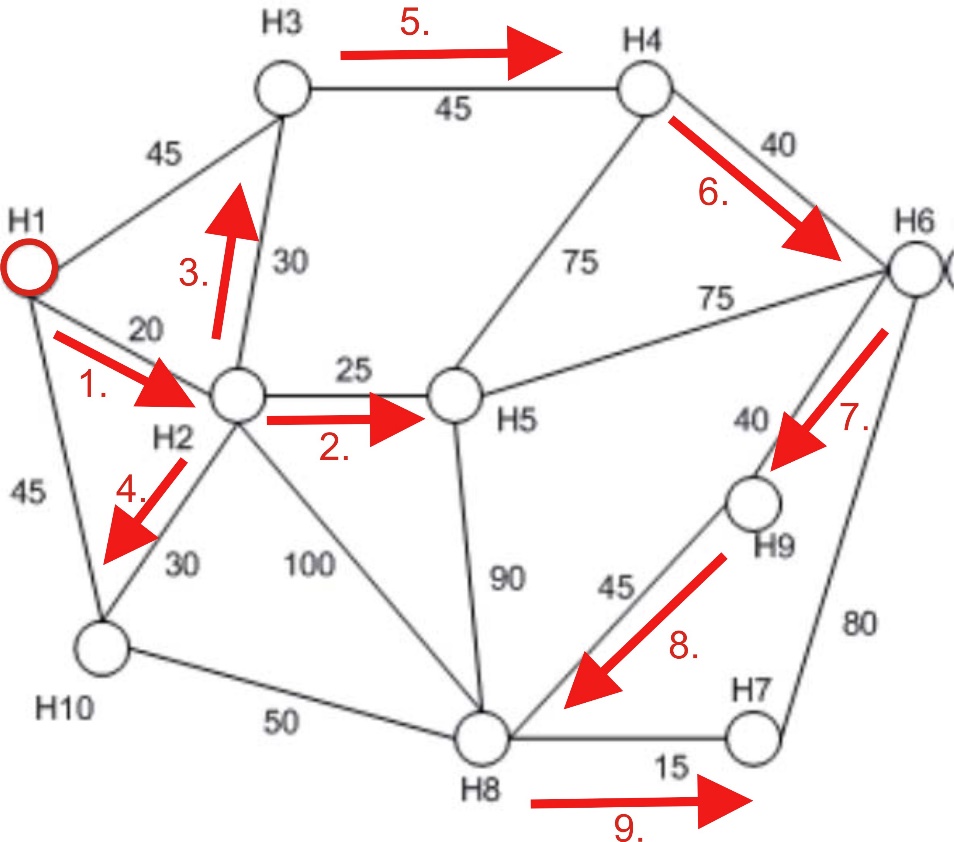
**}** // end while

**}**   
  
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)
* 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).

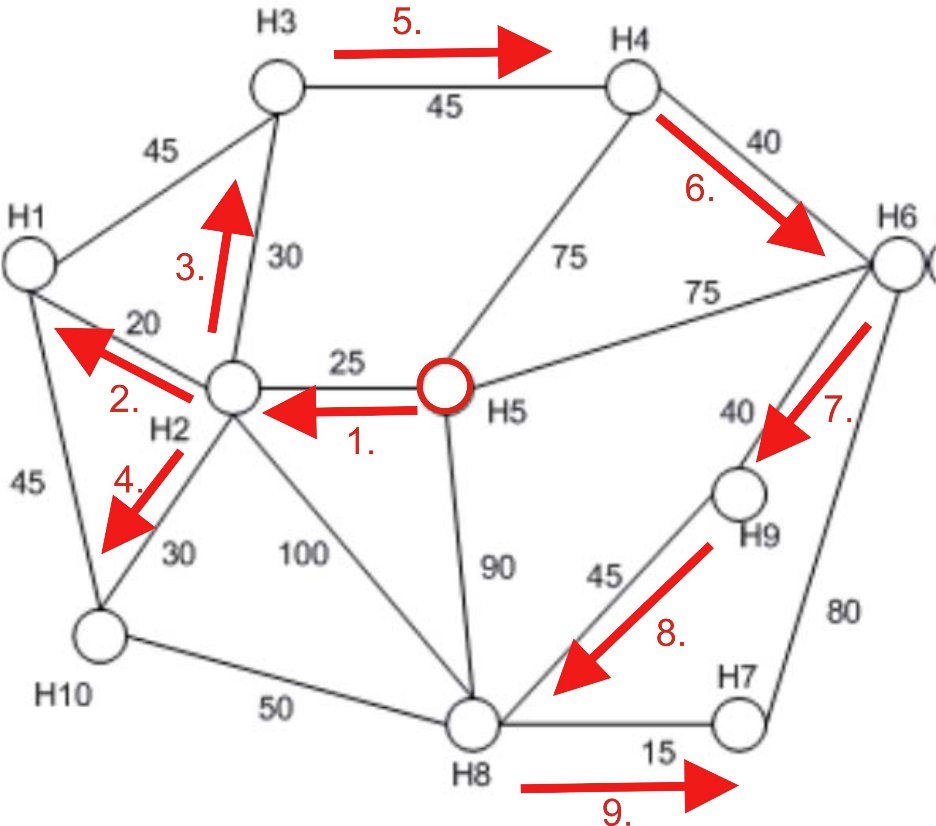












* More than one solution may exist. In this case how many solutions can you find?
* Compare the minimum spanning tree algorithm with the Dijkstra shortest path algorithm   
  and reflect on the principal similarities and differences.
* A graph may be implemented as an adjacency matrix, or an adjacency list. What is the complexity (in big O notation) of the minimum spanning tree algorithm with the two different graph models?