# Title

Graphs:  shortest path

## Perspective

Modeling problems from the real physical world will in many cases call for a graph, such as:

* Find the cheapest plane ticket from Copenhagen to Rome.
* Find the quickest route from Lautrupvang to Tivoli by bike.
* Determine the cheapest way to supply pipes for fresh water to houses under construction.
* Find the most efficient route through a complex data network.
* In a speech recognition system find the most likely word in a stream of spoken words.

The choice of implementation of a graph together with associated algorithms is highly dependent on the problem to be solved. Some attempts have been made to supply graph libraries to toolboxes of different compilers, but they are often over‐killed with functionality and very often cannot satisfy your actual needs.

Answer: do it yourselves!

Knowledge of some very basic graph‐searching algorithms is mandatory. They are fundamental to most of the more advanced algorithms.  
  
Assignment

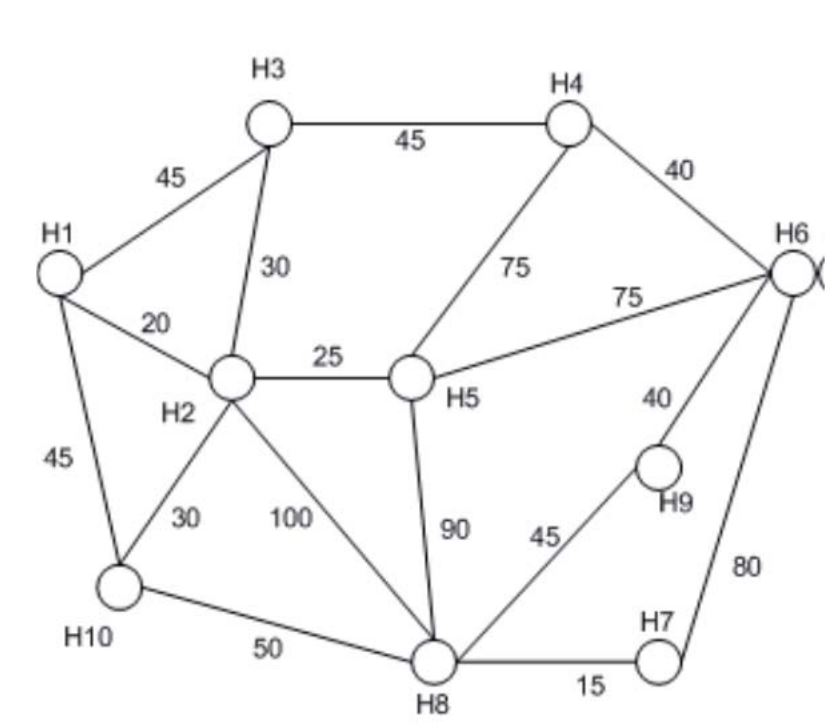


Figure 1- Connection graph (e.g. connections between houses or transport stops)

Run Dijkstra’s Shortest Path algorithm by hand on the figure above.

* Use H1 and H5 as two different source (starting) vertices (and others, if you wish).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Start H1*** | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 |
| Dist | **0** |  |  |  |  |  |  |  |  |  |
| Pred |  |  |  |  |  |  |  |  |  |  |
| Dist |  | **20** | 45 |  |  |  |  |  |  | 45 |
| Pred |  | **H1** | H1 |  |  |  |  |  |  | H1 |
| Dist |  |  | **45** |  | 45 |  |  | 120 |  | 45 |
| Pred |  |  | **H1** |  | H2 |  |  | H2 |  | H1 |
| Dist |  |  |  | 90 | **45** |  |  | 120 |  | 45 |
| Pred |  |  |  | H3 | **H2** |  |  | H2 |  | H1 |
| Dist |  |  |  | 90 |  | 120 |  | 120 |  | **45** |
| Pred |  |  |  | H3 |  | H5 |  | H2 |  | **H1** |
| Dist |  |  |  | **90** |  | 120 |  | 95 |  |  |
| Pred |  |  |  | **H3** |  | H5 |  | H10 |  |  |
| Dist |  |  |  |  |  | 120 |  | **95** |  |  |
| Pred |  |  |  |  |  | H5 |  | **H10** |  |  |
| Dist |  |  |  |  |  | 120 | **110** |  | 140 |  |
| Pred |  |  |  |  |  | H5 | **H8** |  | H8 |  |
| Dist |  |  |  |  |  | **120** |  |  | 140 |  |
| Pred |  |  |  |  |  | **H5** |  |  | H8 |  |
| Dist |  |  |  |  |  |  |  |  | **140** |  |
| Pred |  |  |  |  |  |  |  |  | **H8** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Goal | Cost | Path (List of vertices) |  |  |
| H1 | 0 |  |  |  |
| H2 | 20 | H1 |  |  |
| H3 | 45 | H1 |  |  |
| H4 | 90 | H3 | H1 |  |
| H5 | 45 | H2 | H1 |  |
| H6 | 120 | H5 | H2 | H1 |
| H7 | 110 | H8 | H10 | H1 |
| H8 | 95 | H10 | H1 |  |
| H9 | 140 | H8 | H10 | H1 |
| H10 | 45 | H1 |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Start/Source H5*** | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 |
| Dist |  |  |  |  | **0** |  |  |  |  |  |
| Pred |  |  |  |  |  |  |  |  |  |  |
| dist |  | **25** |  | 75 |  | 75 |  | 90 |  |  |
| pred |  | **H5** |  | H5 |  | H5 |  | H5 |  |  |
| dist | **45** |  | 55 | 75 |  | 75 |  | 90 |  | 55 |
| pred | **H2** |  | H2 | H5 |  | H5 |  | H5 |  | H2 |
| dist |  |  | **55** | 75 |  | 75 |  | 90 |  | 55 |
| pred |  |  | **H2** | H5 |  | H5 |  | H5 |  | H2 |
| dist |  |  |  | 75 |  | 75 |  | 90 |  | **55** |
| pred |  |  |  | H5 |  | H5 |  | H5 |  | **H2** |
| dist |  |  |  | **75** |  | 75 |  | 90 |  |  |
| pred |  |  |  | **H5** |  | H5 |  | H5 |  |  |
| dist |  |  |  |  |  | **75** |  | 90 |  |  |
| pred |  |  |  |  |  | **H5** |  | H5 |  |  |
| dist |  |  |  |  |  |  | 155 | **90** | 115 |  |
| pred |  |  |  |  |  |  | H6 | **H5** | H6 |  |
| dist |  |  |  |  |  |  | **105** |  | 115 |  |
| pred |  |  |  |  |  |  | **H8** |  | H6 |  |
| dist |  |  |  |  |  |  |  |  | **115** |  |
| pred |  |  |  |  |  |  |  |  | **H6** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Goal | Cost | Path (List of vertices) |  |  |
| H1 | 45 | H2 | H5 |  |
| H2 | 25 | H5 |  |  |
| H3 | 55 | H2 | H5 |  |
| H4 | 75 | H5 |  |  |
| H5 | 0 |  |  |  |
| H6 | 75 | H5 |  |  |
| H7 | 105 | H8 | H5 |  |
| H8 | 90 | H5 |  |  |
| H9 | 115 | H6 | H5 |  |
| H10 | 55 | H2 | H5 |  |

* The connection between H8 and H10 is now disconnected.   
  Run the algorithm again with H1 as the source vertex.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (no connection between H8 and H10) | | | |  |  |  |  |  |  |  |
| ***Start H1*** | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 |
| Dist | **0** |  |  |  |  |  |  |  |  |  |
| Pred |  |  |  |  |  |  |  |  |  |  |
| Dist |  | **20** | 45 |  |  |  |  |  |  | 45 |
| Pred |  | **H1** | H1 |  |  |  |  |  |  | H1 |
| Dist |  |  | **45** |  | 45 |  |  | 120 |  | 45 |
| Pred |  |  | **H1** |  | H2 |  |  | H2 |  | H1 |
| Dist |  |  |  | 90 | **45** |  |  | 120 |  | 45 |
| Pred |  |  |  | H3 | **H2** |  |  | H2 |  | H1 |
| Dist |  |  |  | 90 |  | 120 |  | 120 |  | **45** |
| Pred |  |  |  | H3 |  | H5 |  | H2 |  | **H1** |
| Dist |  |  |  | **90** |  | 120 |  | 120 |  |  |
| Pred |  |  |  | **H3** |  | H5 |  | H2 |  |  |
| Dist |  |  |  |  |  | **120** |  | 120 |  |  |
| Pred |  |  |  |  |  | **H5** |  | H2 |  |  |
| Dist |  |  |  |  |  |  | 200 | **120** | 130 |  |
| Pred |  |  |  |  |  |  | H6 | **H2** | H6 |  |
| Dist |  |  |  |  |  |  | 135 |  | **130** |  |
| Pred |  |  |  |  |  |  | H8 |  | **H6** |  |
| Dist |  |  |  |  |  |  | **135** |  |  |  |
| Pred |  |  |  |  |  |  | **H8** |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Goal | Cost | Path (List of vertices) |  |  |  |
| H1 | 0 |  |  |  |  |
| H2 | 20 | H1 |  |  |  |
| H3 | 45 | H1 |  |  |  |
| H4 | 90 | H3 | H1 |  |  |
| H5 | 45 | H2 | H1 |  |  |
| H6 | 120 | H5 | H2 | H1 |  |
| H7 | 135 | H8 | H2 | H1 |  |
| H8 | 120 | H2 | H1 |  |  |
| H9 | 130 | H6 | H5 | H2 | H1 |
| H10 | 45 | H1 |  |  |  |

* What is the complexity (in big‐O notation) of the Dijkstra algorithm?    
  Hint: The implementation of the priority queue may influence your answer.

O(|E|+|V|\*log|V|)