

Graphs

Irina Prosvirnina

- The shortest-path problem
- Dijkstra's algorithm

Weighted graphs

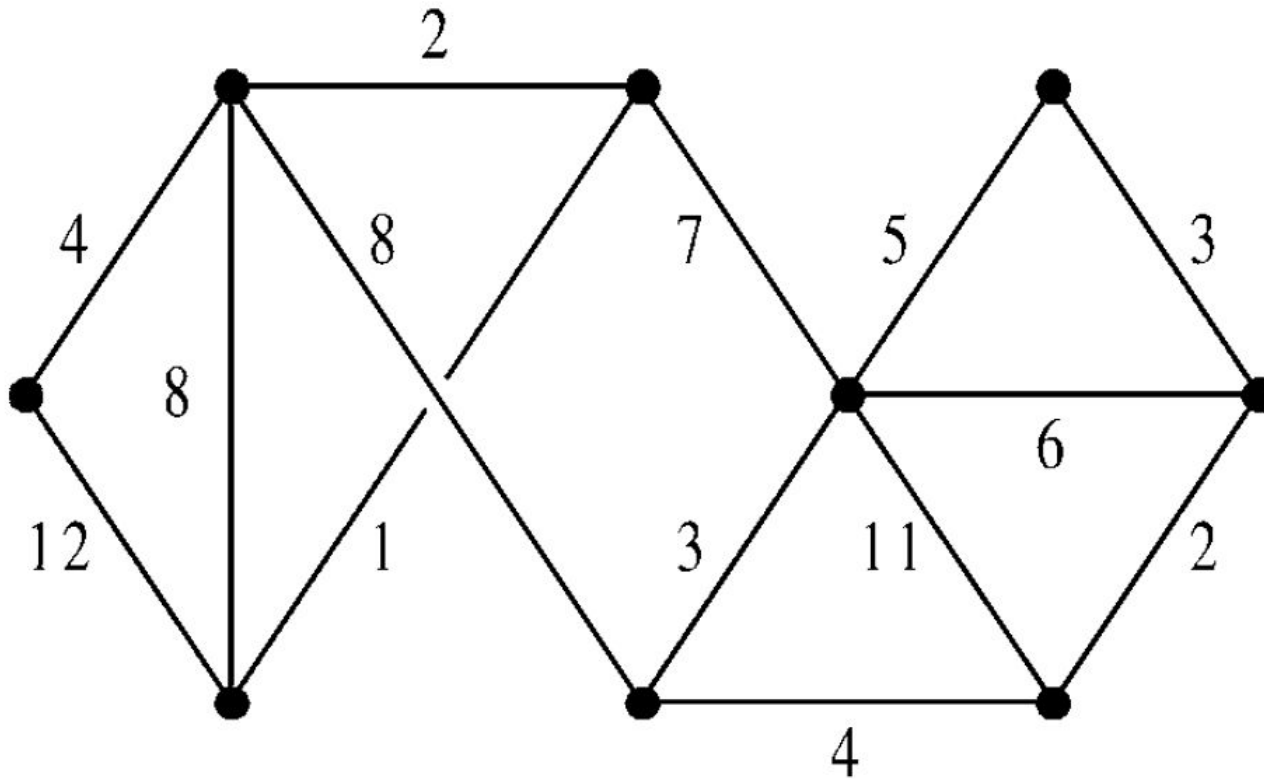
Definition 5

A **weighted graph** is a graph Γ together with a function $w: E_\Gamma \rightarrow \mathbb{R}^+ \cup \{0\}$.

If e is an edge of Γ , then the number $w(e)$ is called the **weight** of e .

Weighted graphs

For example, the figure is a diagram of a weighted graph.



Weighted graphs

Definition 1

Let Γ be a weighted graph.

For any subgraph Γ' of Γ we define the **weight** of Γ' , $w(\Gamma')$, to be the sum of the weights of its edges.

Symbolically,

$$w(\Gamma') = \sum_{e \in E_{\Gamma'}} w(e).$$

The shortest-path problem

Let Γ be a connected weighted graph and let v, v' be two vertices of Γ . The **shortest-path problem** is to find a path joining v and v' with the smallest weight.

Of course, since a path is a subgraph of Γ , its weight is defined as the sum of the weights of its edges.

As we are dealing with finite graphs, it should be obvious that a shortest path exists, although there may be more than one shortest path joining a given pair of vertices.

The shortest-path problem

There are various methods for finding a shortest path between two given vertices.

We describe an algorithm which, like the minimal spanning tree algorithm of the previous section, constructs the path one edge at a time.

The shortest-path problem

The **idea** is to begin at the vertex v and move through the graph assigning a number $L(u)$ to each vertex u in turn which represents the length of the shortest path yet discovered from v to u .

These 'length numbers' $L(u)$ are initially considered temporary and may subsequently be changed if we discover a path from v to u which has length less than the currently assigned value $L(u)$.

The shortest-path problem

The algorithm is detailed more precisely below.

It actually constructs a subtree of the graph containing the vertices v and v' ; a shortest path between the two vertices is then the unique path in this tree joining them.

Note that the subtree constructed by this algorithm need not be a spanning subtree of the graph – the algorithm stops as soon as a shortest path joining v and v' has been found.)

Dijkstra's algorithm for finding a shortest path between two given vertices

1. First assign $L(v) = 0$ to the starting vertex v . We say that the vertex v has been labelled with the value 0. Furthermore, this label is permanent as we will not subsequently change its value. Since we are constructing a sequence of trees, we also begin with the tree consisting of vertex v only and no edges.

Dijkstra's algorithm for finding a shortest path between two given vertices

2. Let u be the vertex which has most recently been given a permanent label. (Initially $v = u$ as this is the only vertex with a permanent label.) Consider each vertex u' adjacent to u in turn, and give it a temporary label as follows. (Only those vertices u' without a permanent label are considered.)
 - a) If u' is unlabelled, then set $L(u')$ equal to $L(u) + w(e)$ where e is the edge joining u to u' . (If there is more than one such edge e , choose the one with the smallest weight.)

Dijkstra's algorithm for finding a shortest path between two given vertices

- b) If u' is already labelled, then again calculate $L(u) + w(e)$ as above and if this is less than the current value of $L(u')$ then change $L(u')$ to $L(u) + w(e)$; otherwise leave $L(u')$ unchanged.

Dijkstra's algorithm for finding a shortest path between two given vertices

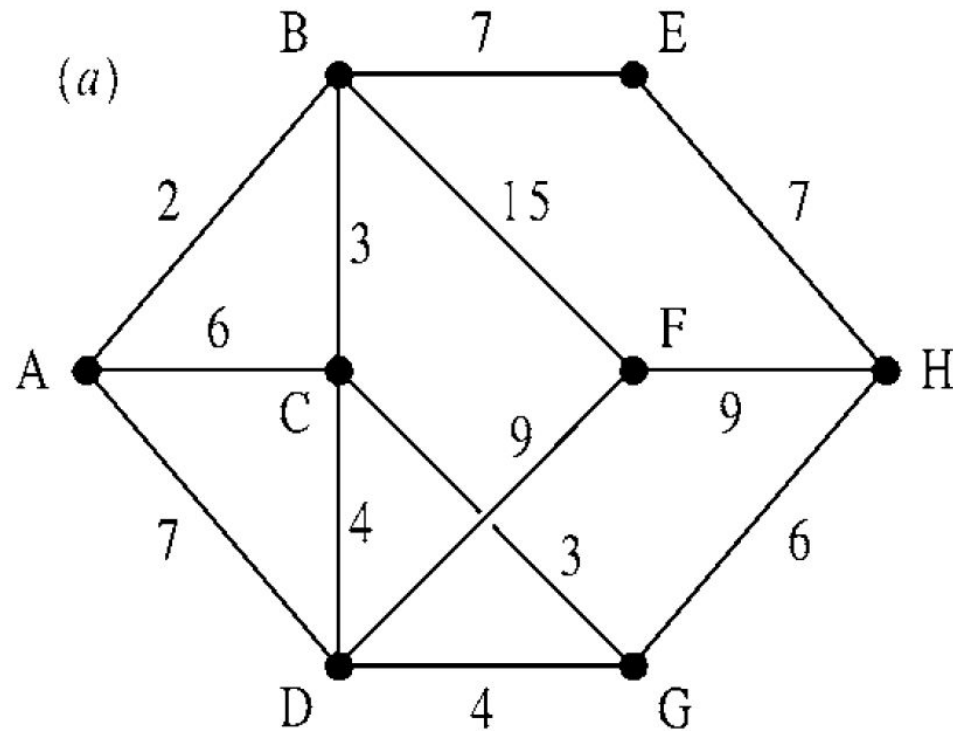
3. Choose a vertex w with the smallest temporary label and make the label permanent. (There may be a choice to be made here as several temporarily labelled vertices may have equal smallest labels. It is also important to realize that w need not be adjacent to u , the most recently labelled vertex.) At the same time adjoin to the tree so far formed the edge which gives rise to the value $L(w)$.

Dijkstra's algorithm for finding a shortest path between two given vertices

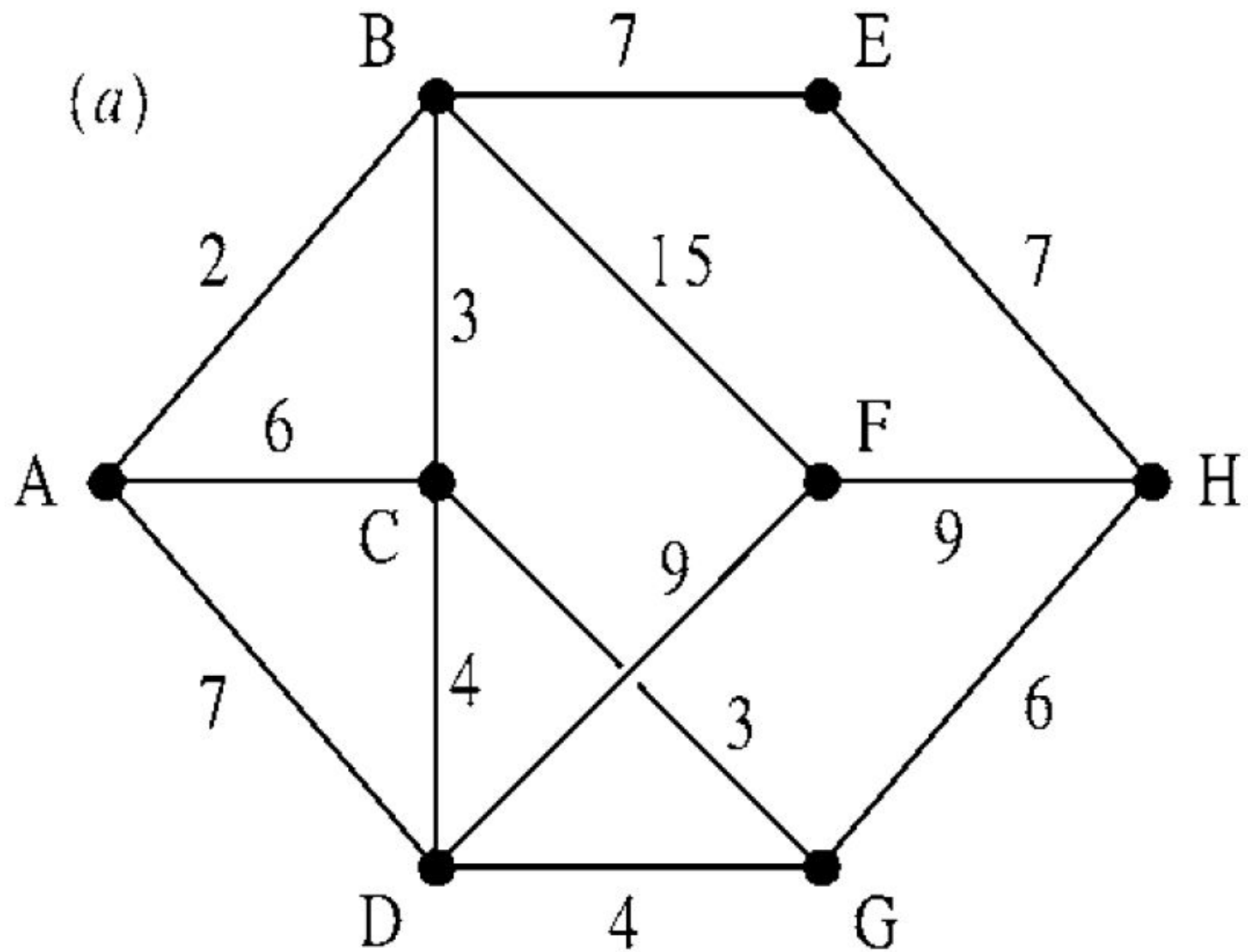
4. Repeat steps 2 and 3 until the final vertex v' has been given a permanent label. The path of shortest length from v to v' is then the unique path in the tree thus formed joining v and v' . Its length is the permanent value of $L(v')$.

Dijkstra's algorithm for finding a shortest path between two given vertices

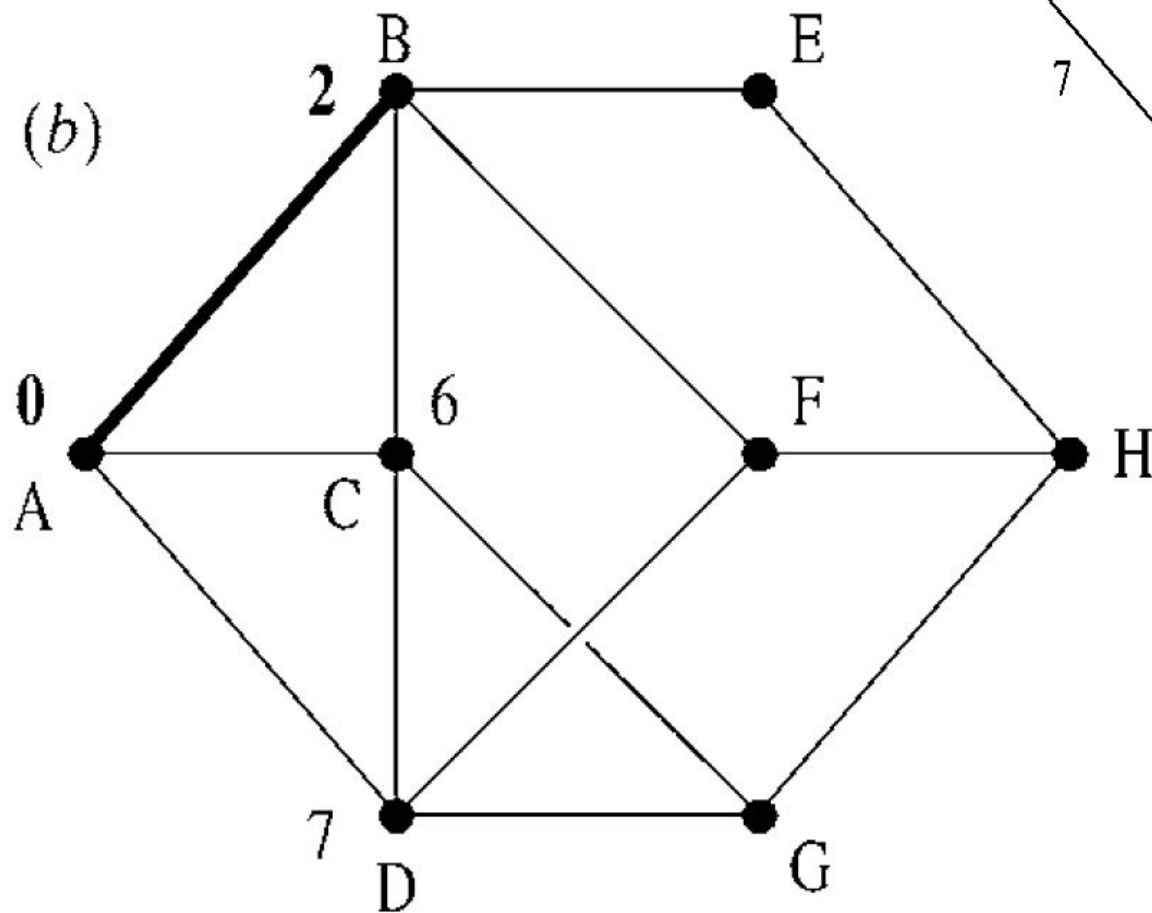
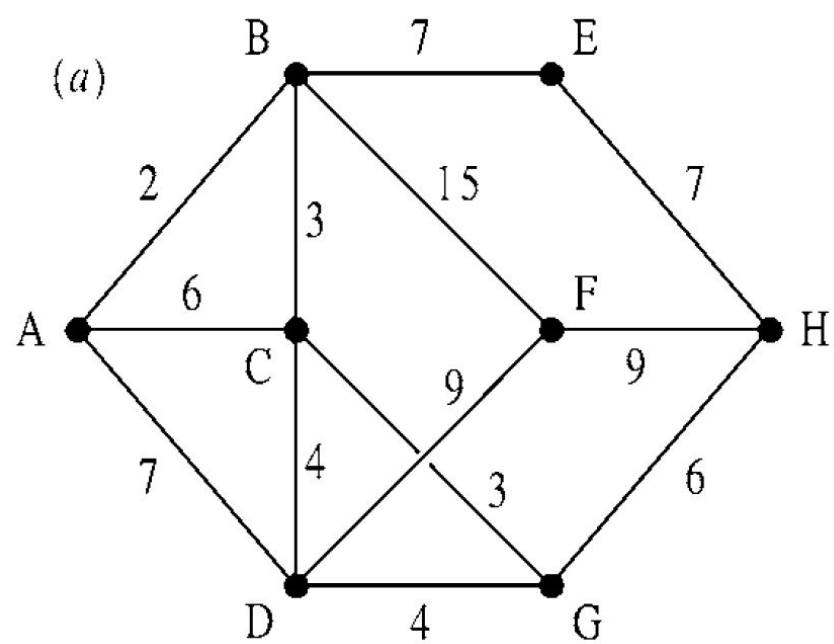
We illustrate this algorithm by constructing a shortest path from A to H in the weighted graph illustrated in the figure.



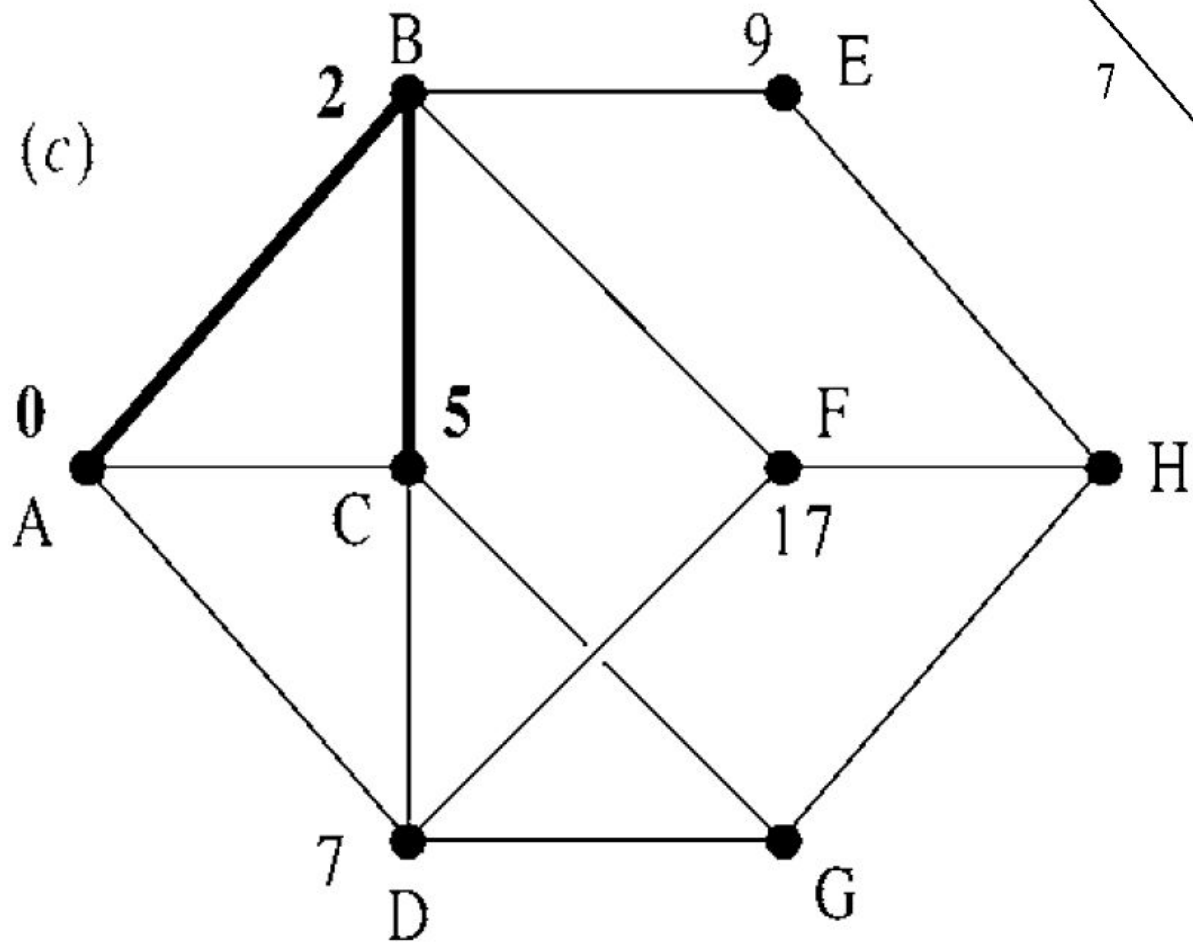
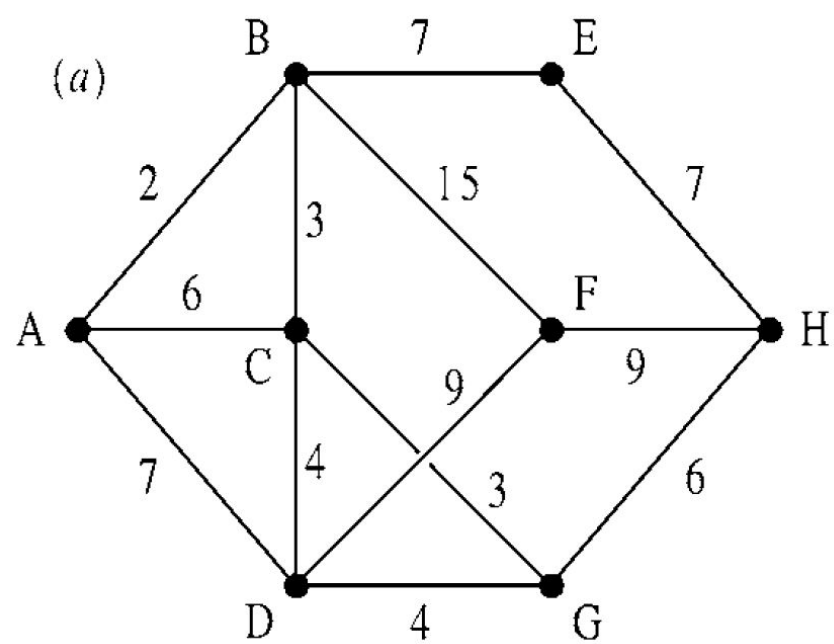
Dijkstra's algorithm



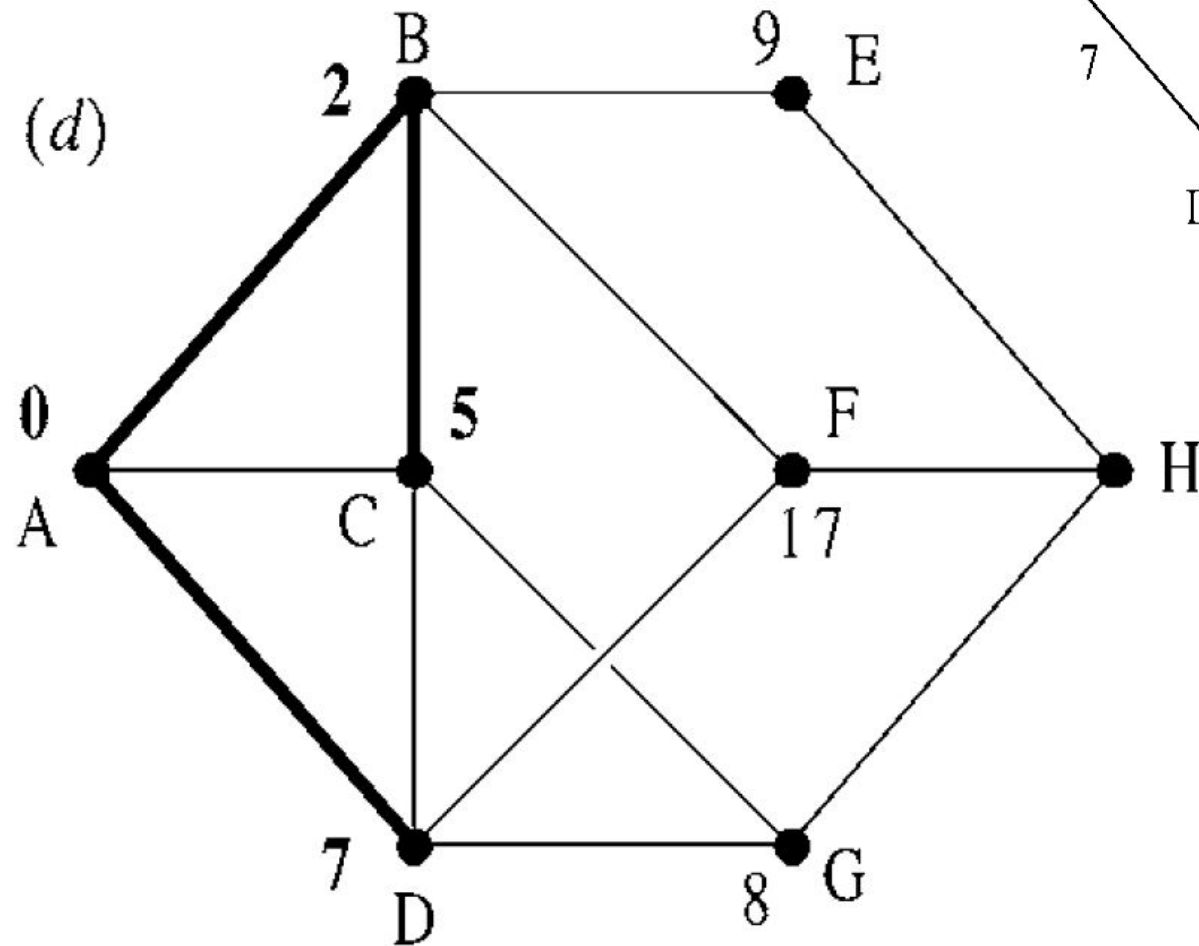
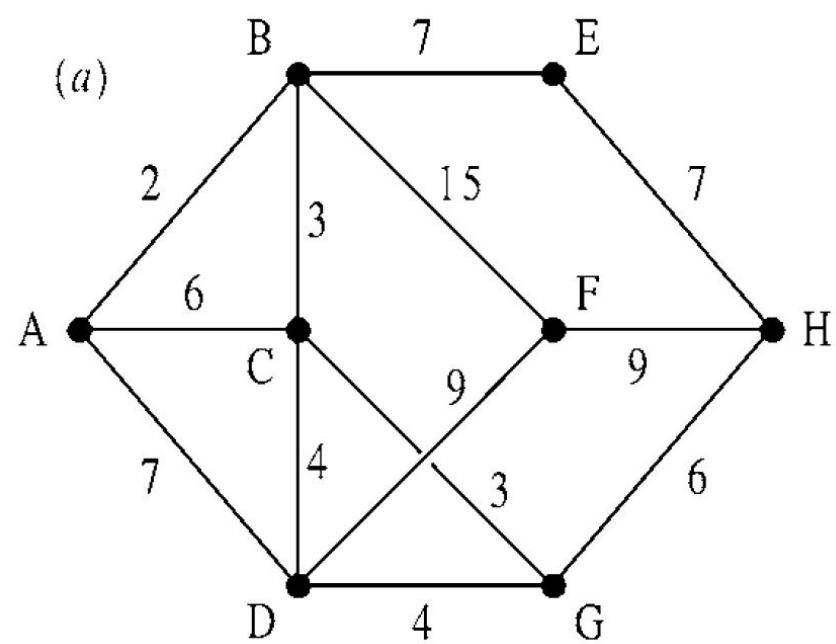
Dijkstra's algorithm



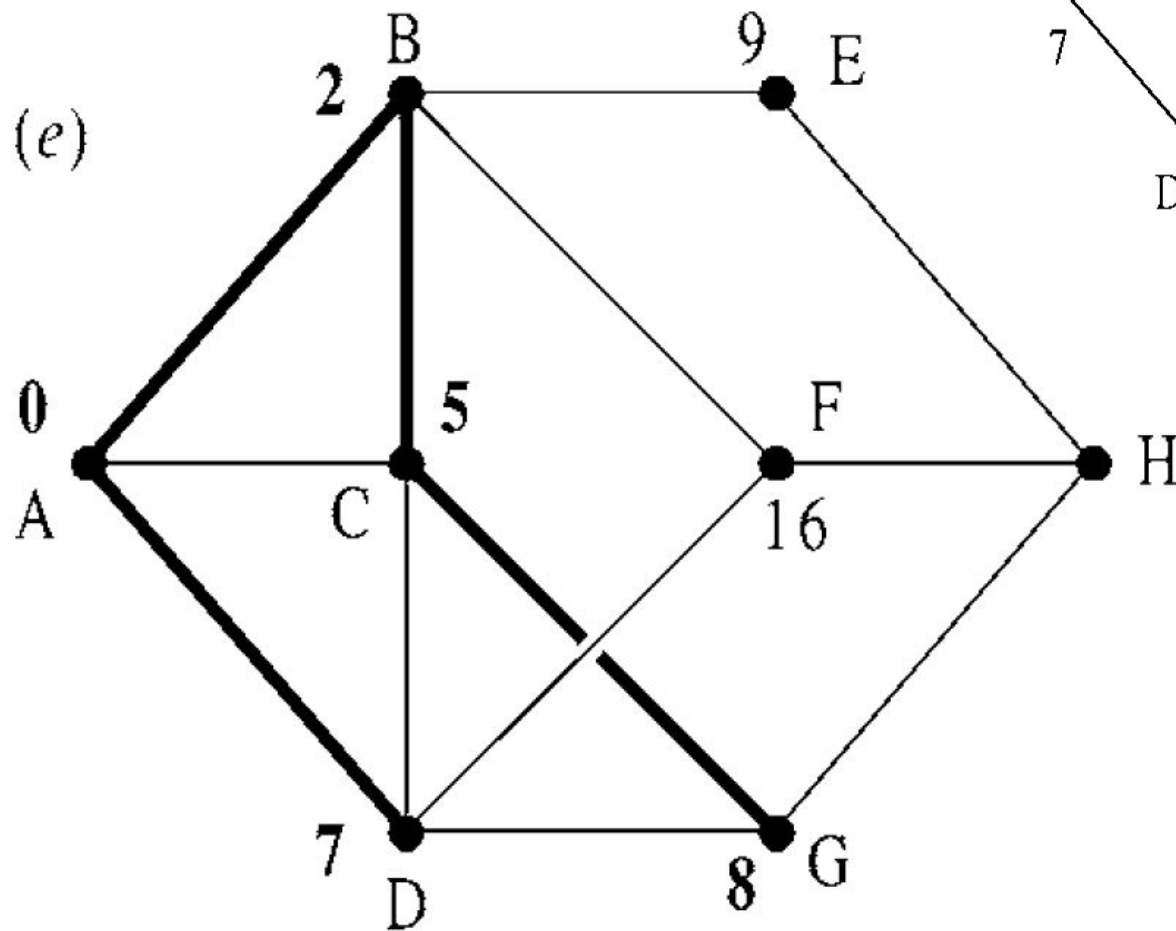
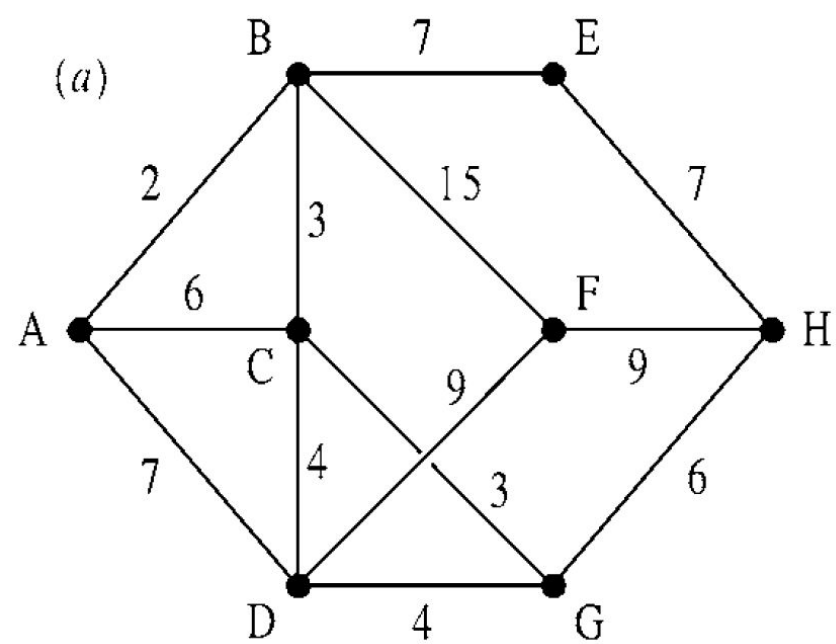
Dijkstra's algorithm



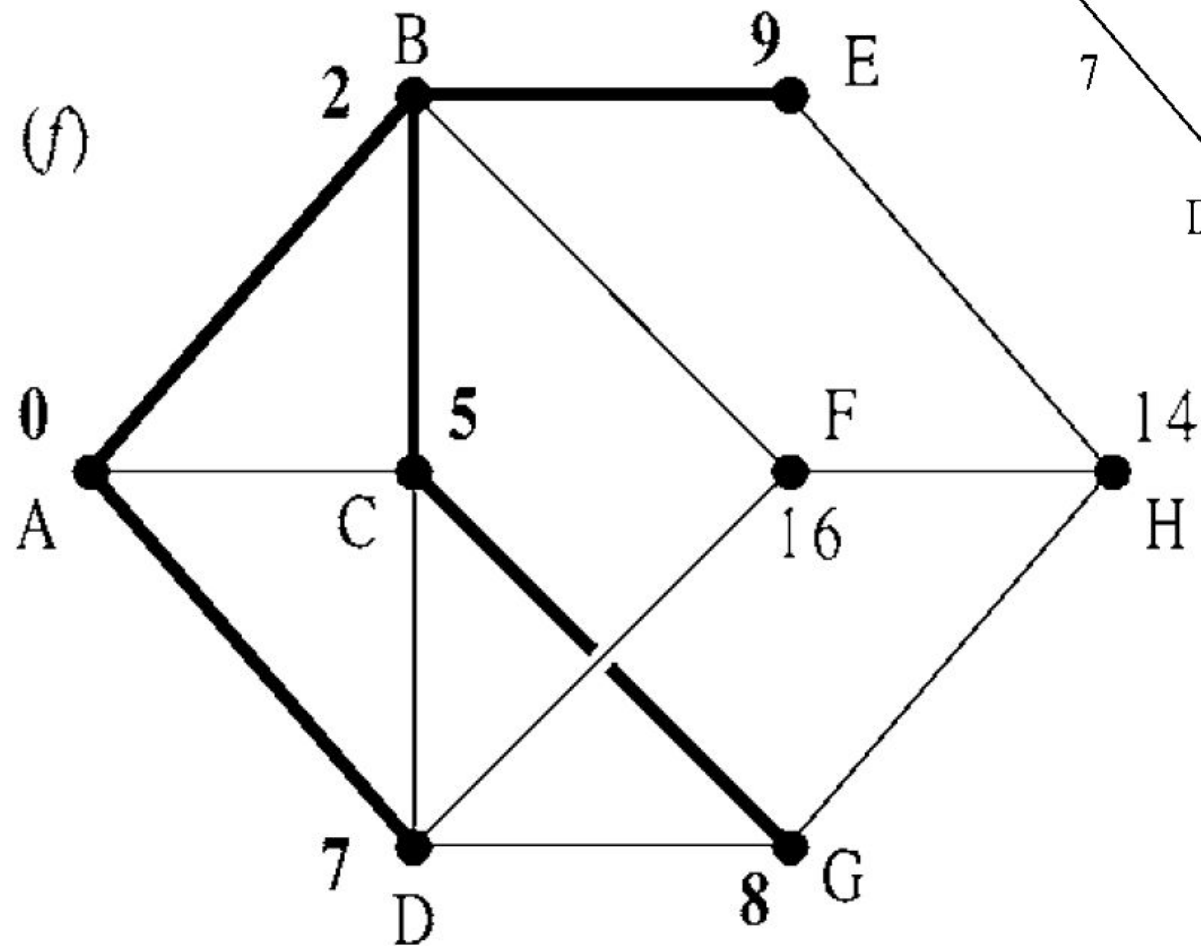
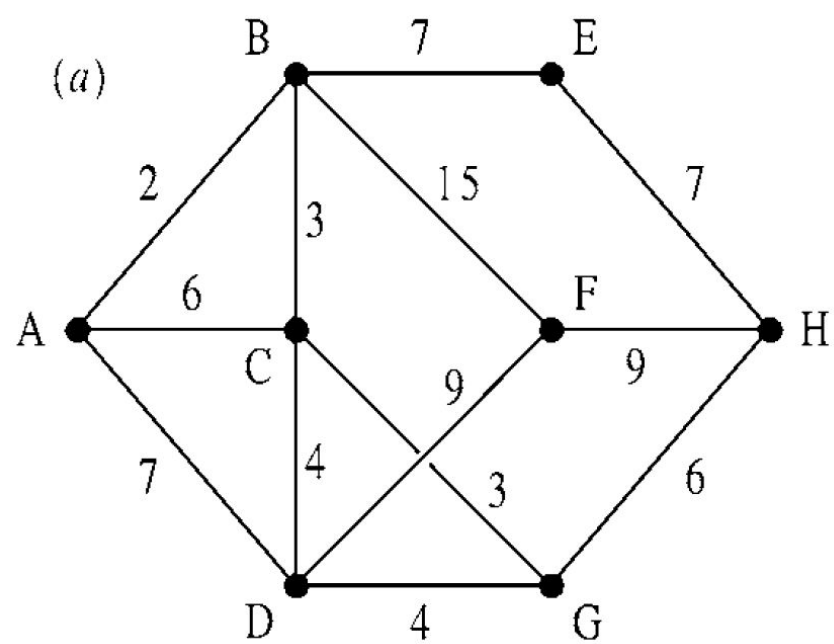
Dijkstra's algorithm



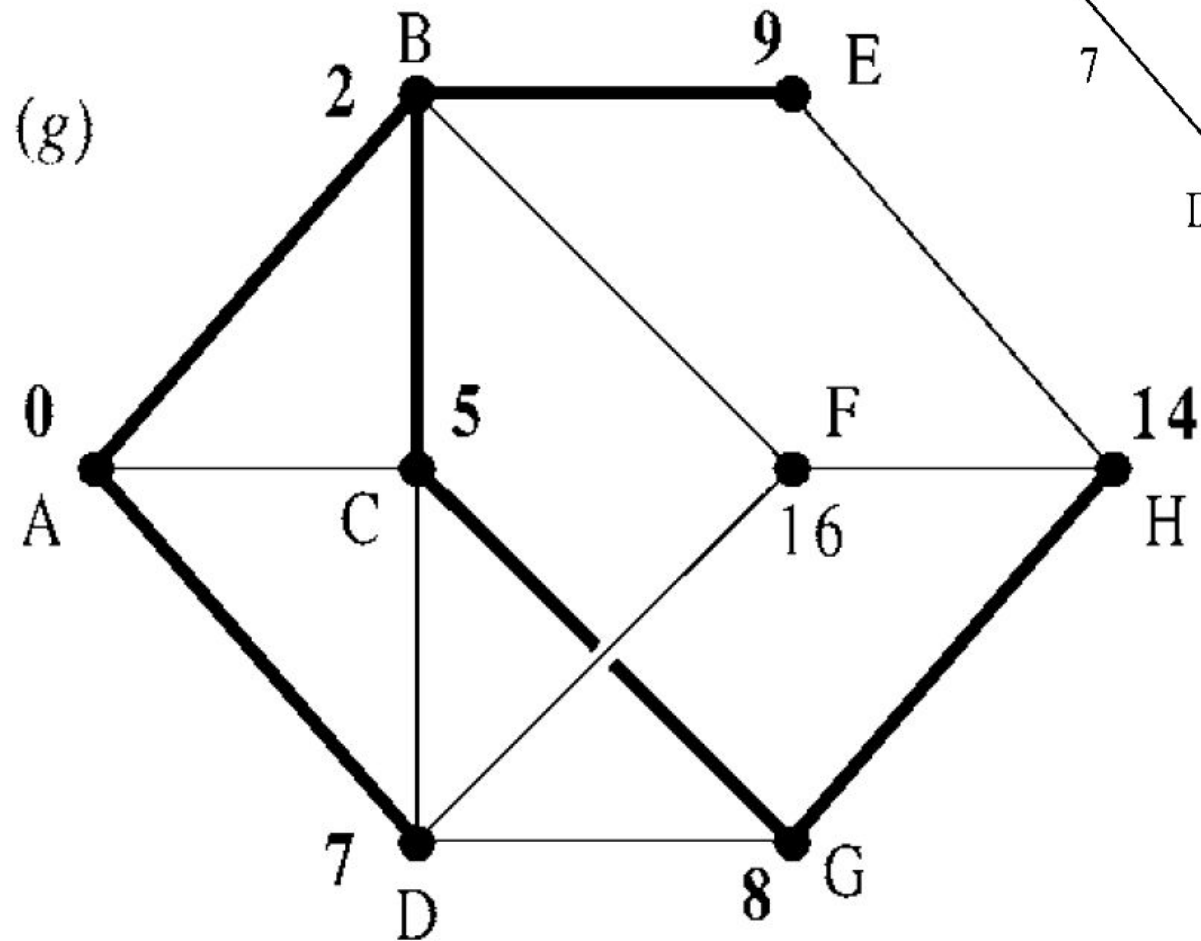
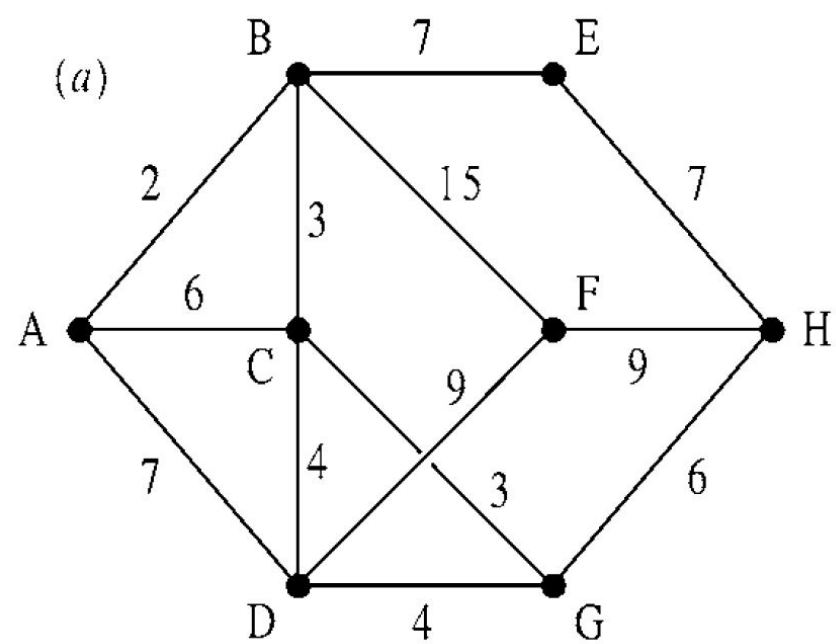
Dijkstra's algorithm



Dijkstra's algorithm

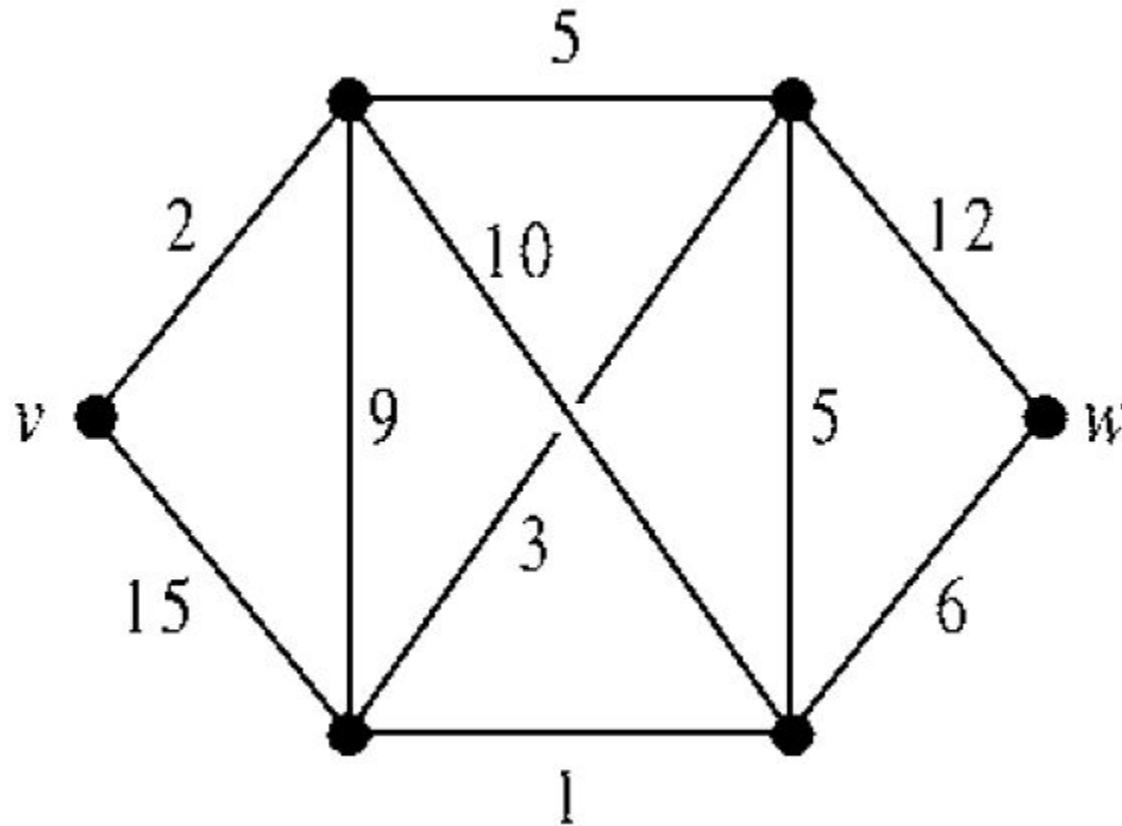


Dijkstra's algorithm



Exercise 2

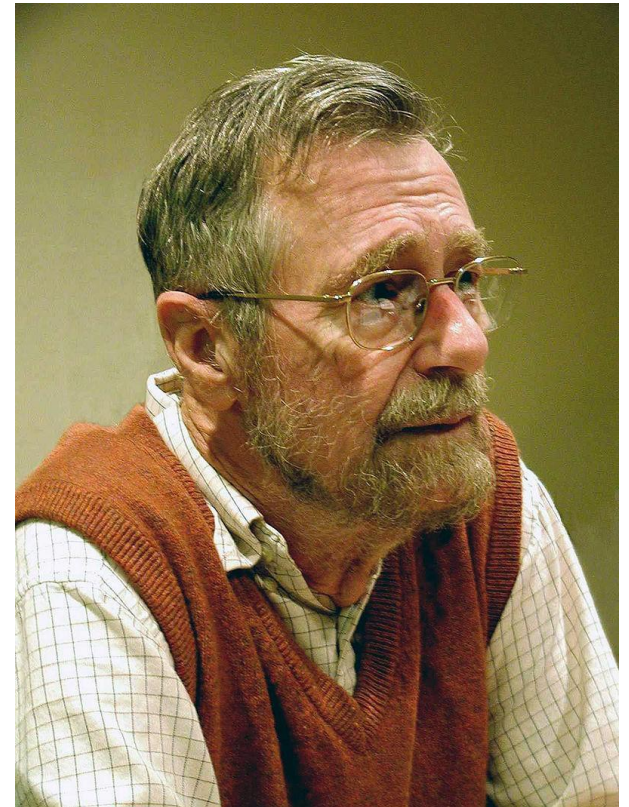
Apply Dijkstra's algorithm to obtain the shortest path from v to w in the following weighted graph.



Biography

Edsger Dijkstra, born in the Netherlands, began programming computers in the early 1950s while studying theoretical physics at the University of Leiden.

In 1952, realizing that he was more interested in programming than in physics, he quickly completed the requirements for his physics degree and began his career as a programmer, even though programming was not a recognized profession.

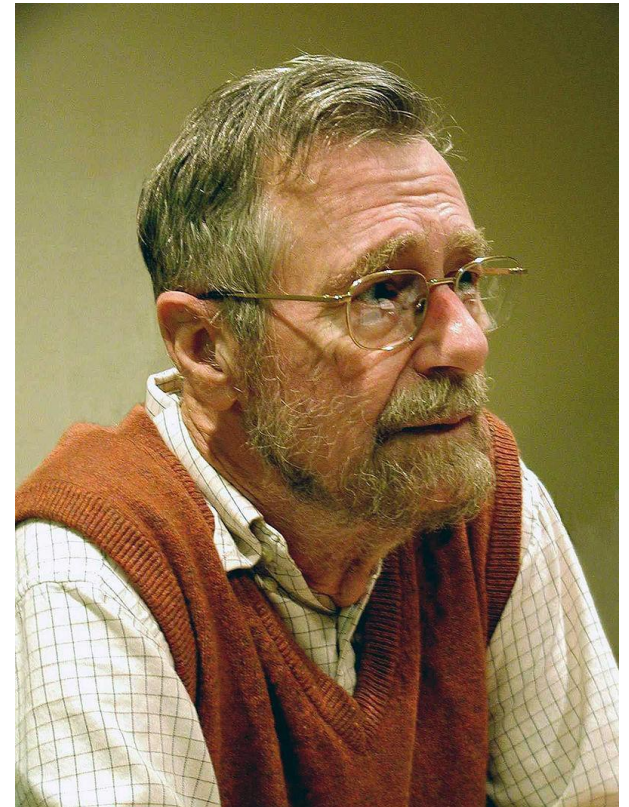


**Edsger Wybe Dijkstra
1930 – 2002**

Biography

In 1957, the authorities in Amsterdam refused to accept “programming” as his profession on his marriage license. However, they did accept “theoretical physicist” when he changed his entry to this.

Dijkstra was one of the most forceful proponents of programming as a scientific discipline.

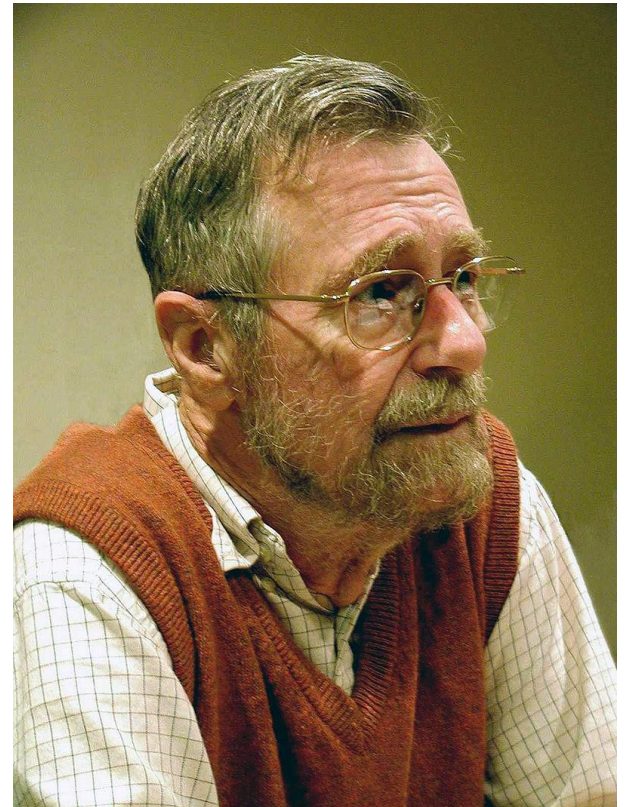


**Edsger Wybe Dijkstra
1930 – 2002**

Biography

In 1972 Dijkstra received the Turing Award from the Association for Computing Machinery, one of the most prestigious awards in computer science.

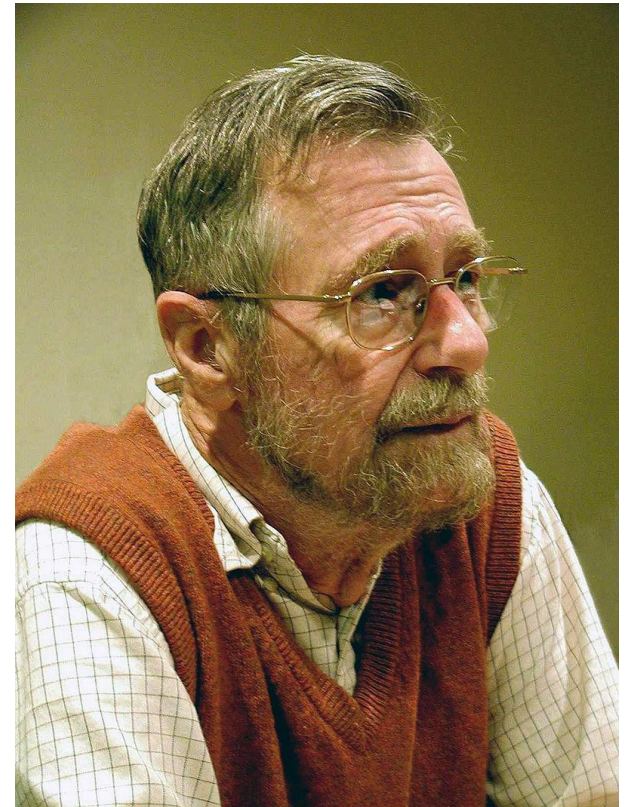
Dijkstra became a Burroughs Research Fellow in 1973, and in 1984 he was appointed to a chair in Computer Science at the University of Texas, Austin.



**Edsger Wybe Dijkstra
1930 – 2002**

Biography

He has made fundamental contributions to the areas of operating systems, including deadlock avoidance; programming languages, including the notion of structured programming; and algorithms.



Edsger Wybe Dijkstra
1930 – 2002