Introduction to Algorithms

CELEN086

Seminar 9
(w/c 09/12/2024)

Outline

In this seminar, we will study and review on following topics:

- Complete graph
- Bipartite graph
- Topological Sorting
- Shortest path and minimum cost
- Dijkstra's algorithm on weighted graph

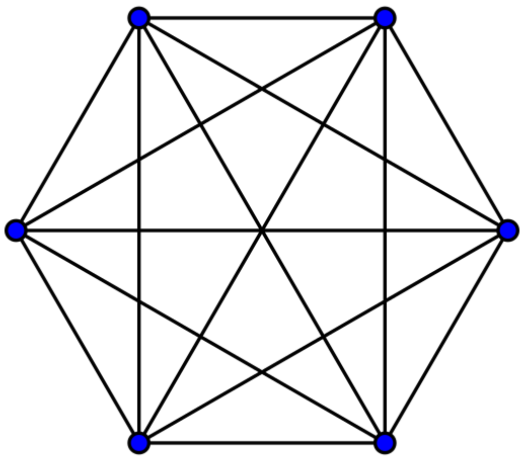
You will also learn useful Math/CS concepts and vocabularies.

Complete graph

Draw a complete graph with n vertices, where $n = 6$.

What is the degree of each vertex?

Degree of each vertex: 5



How many edges in total?

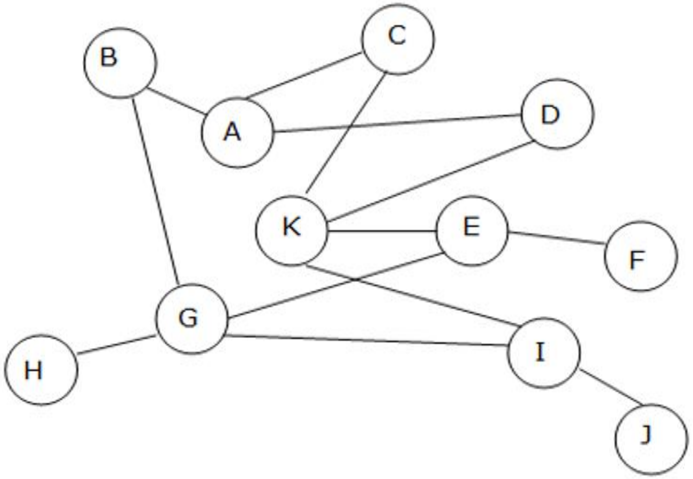
Total number of edges:

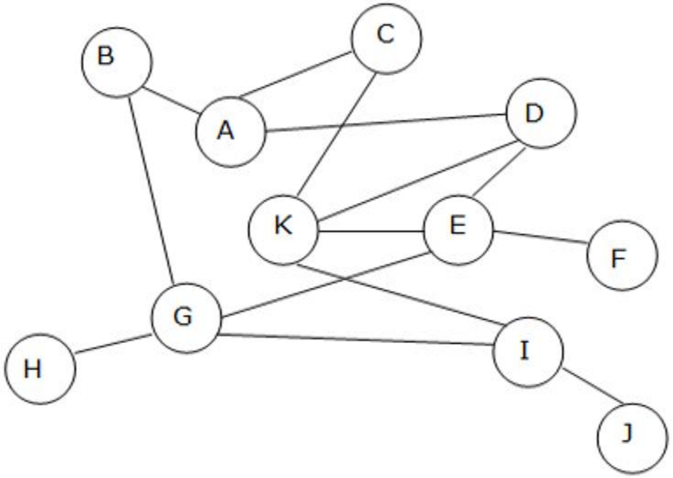
$$\binom{6}{2} = \frac{6 \times 5}{2} = 15$$

Practice: for $n = 7$, answer the same questions.

Bipartite graph

Which one is bipartite?



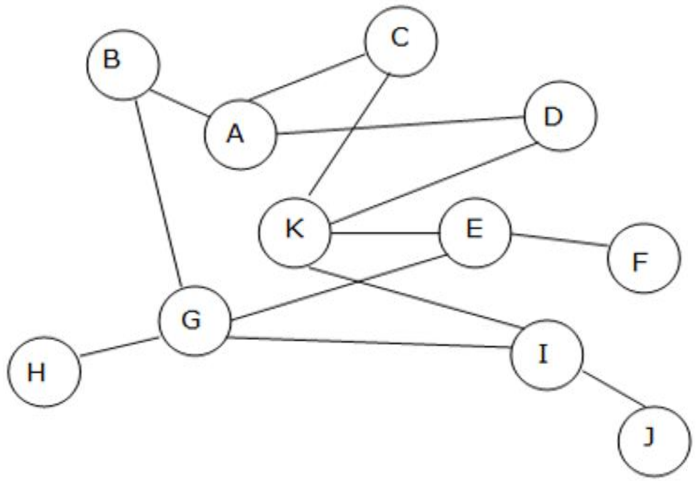


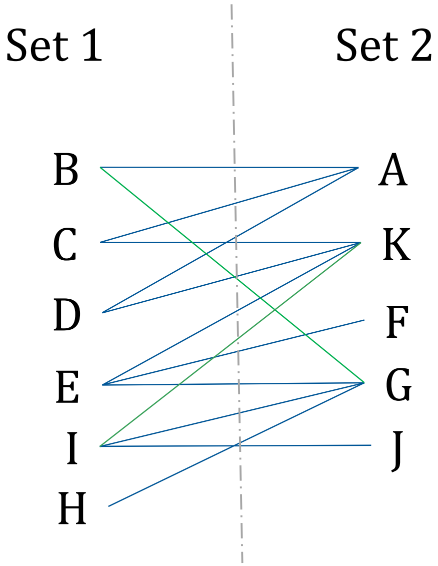
Not bipartite

Because there is a triangular cycle K-D-E-K.

Practice

For this bipartite graph, separate vertices into two sets and draw an equivalent graph.



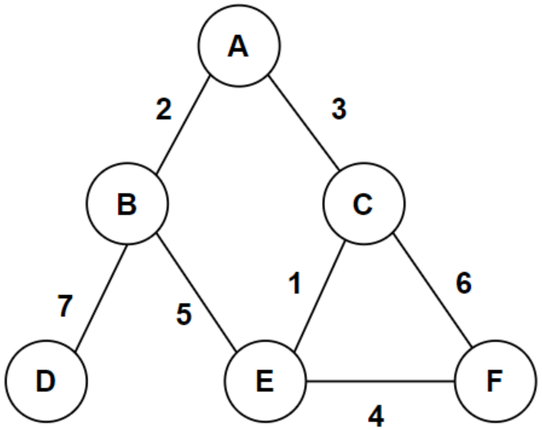


Note:
Double check number of vertices and degrees of each vertex.

Shortest path and minimum cost

Find all the paths between A and F.

Then compute the cost of each path.



A-B-E-F	2+5+4=11
A-B-E-C-F	2+5+1+6=14
A-C-F	3+6=9
A-C-E-F	3+1+4=8

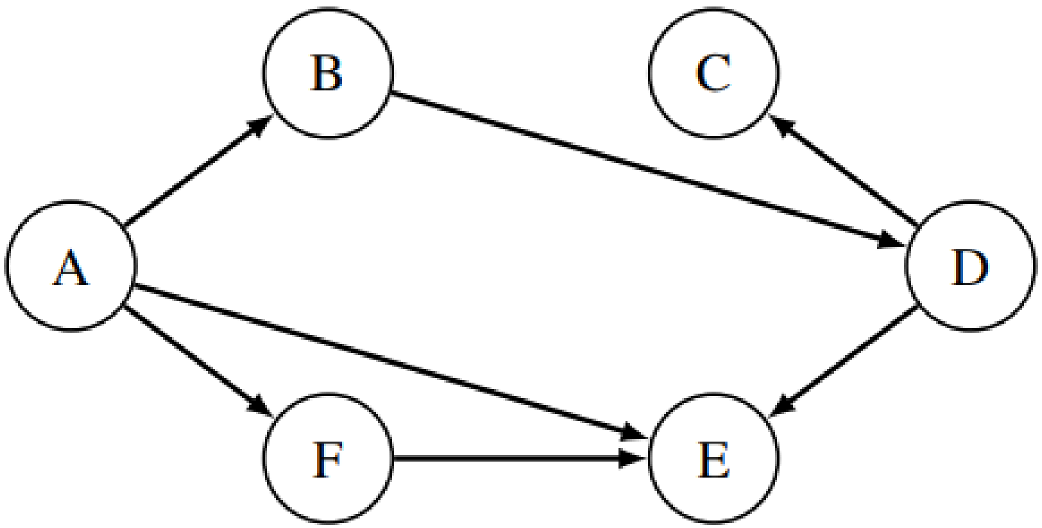
Find the shortest path between A and F, and the minimum cost.

Shortest path between A and F is A-C-E-F,
with minimum cost 8.

Note:
It may happen that there is more than one shortest path, but the minimum cost remains the same.

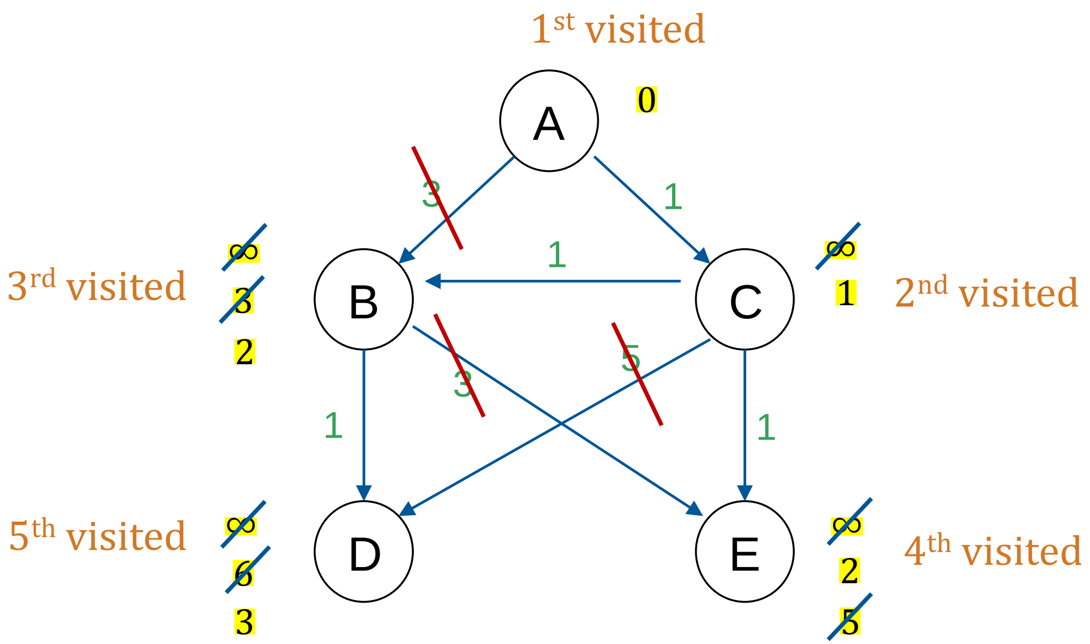
Topological Sorting

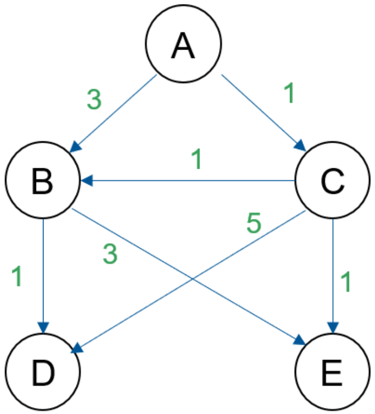
Give a valid topological ordering of the graph. Is it unique?



Dijkstra’s algorithm

Find shortest paths and minimum costs from A to other vertices.





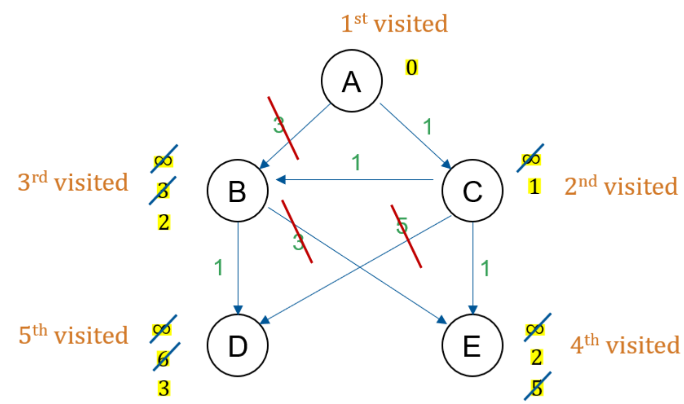
Note:
When selecting next target vertex from unvisited set, choose the one with smallest distance value. If there are more than one possible choices, you may choose either. e.g., choose E first (Lecture 9) or choose B first (Seminar 9).

Dijkstra's algorithm

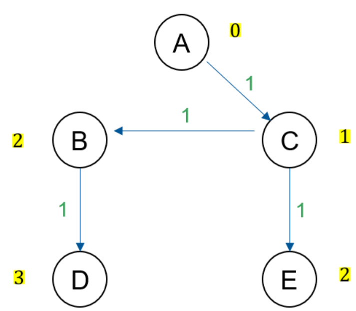
Note about the working process for applying Dijkstra's algorithm.

You may draw a separate graph indicating the process of vertex and edge selections:

- show the order of visited vertices
- show the updated distant values of each vertex along the process (you may cross out old values and edges when you update them)



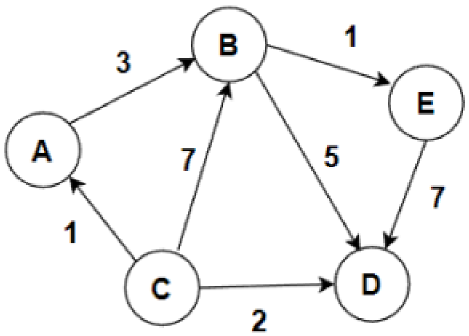
In the end, draw another neat graph and tabulate your final answer:



Vertices	Shortest path	Minimum cost
A to B	A-C-B	2
A to C	A-C	1
A to D	A-C-B-D	3
A to E	A-C-E	2

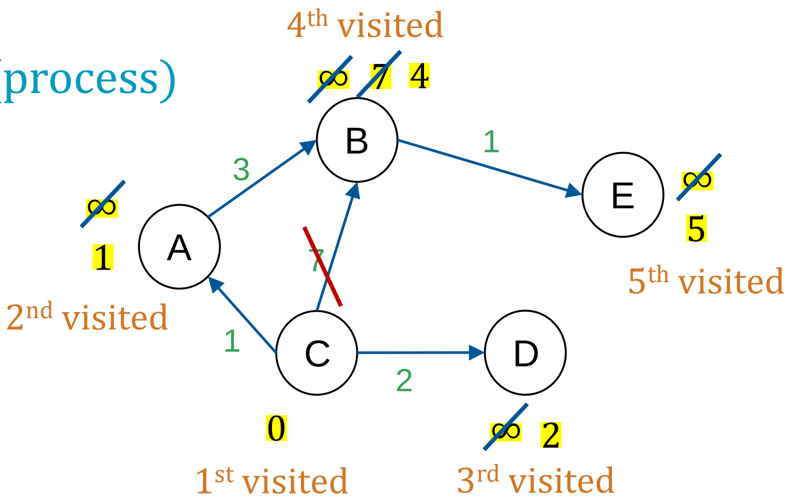
Practice: directed graph

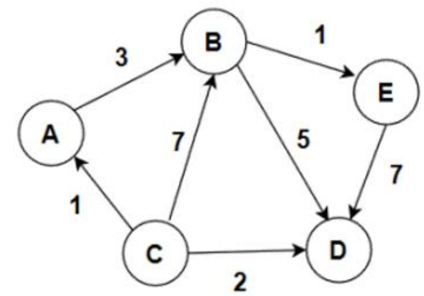
Apply Dijkstra's algorithm to the following weighted and directed graph, to find the shortest paths between Vertex C to all other vertices. Also, state the minimum cost of each path.



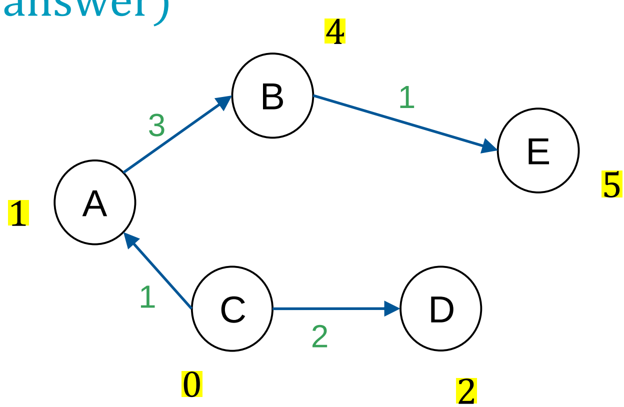
Solution

(process)





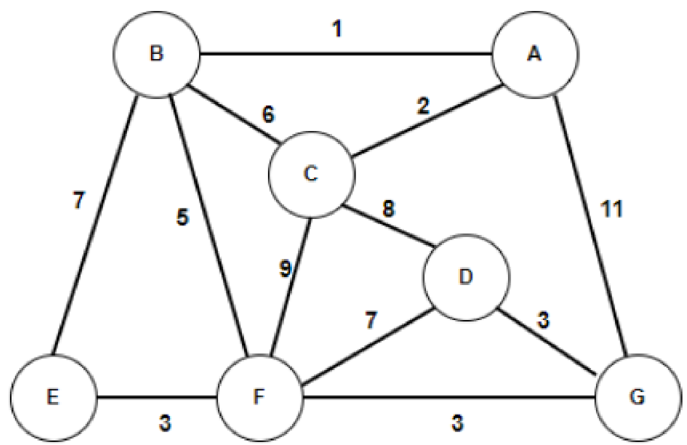
(final answer)



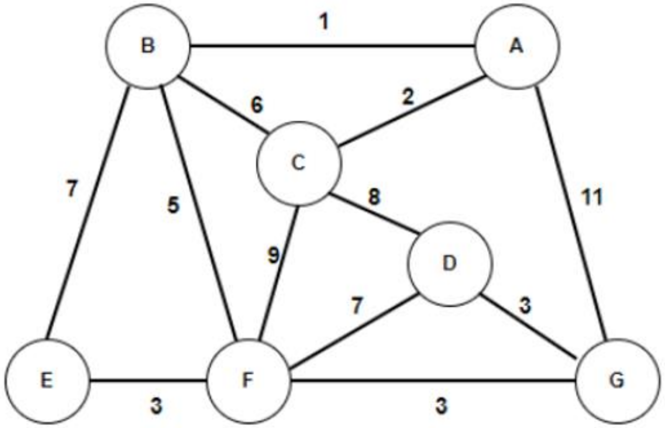
Vertices	Shortest path	Minimum cost
C to A	C-A	1
C to B	C-A-B	4
C to D	C-D	2
C to E	C-A-B-E	5

Practice: undirected graph

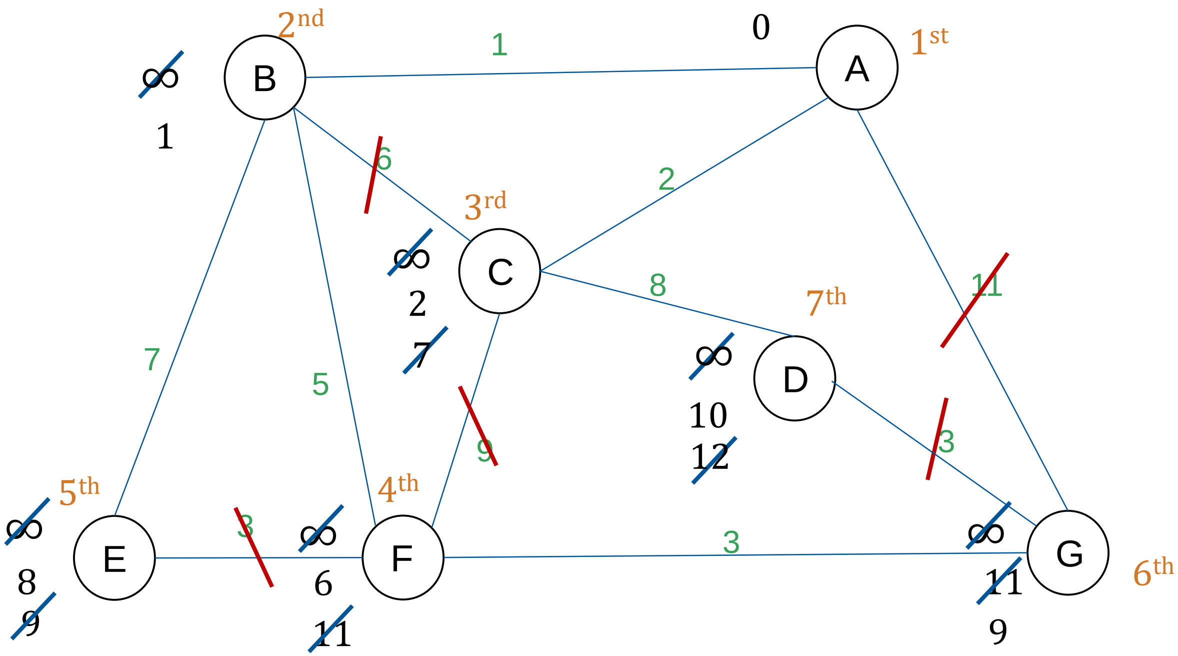
Apply Dijkstra's algorithm to solve question on the weighted and undirected graph (from ILW Q4): Suppose the weights describe distances between places. You are living at place A and would like to figure out shortest paths and distances from A to all the other places.



Solution

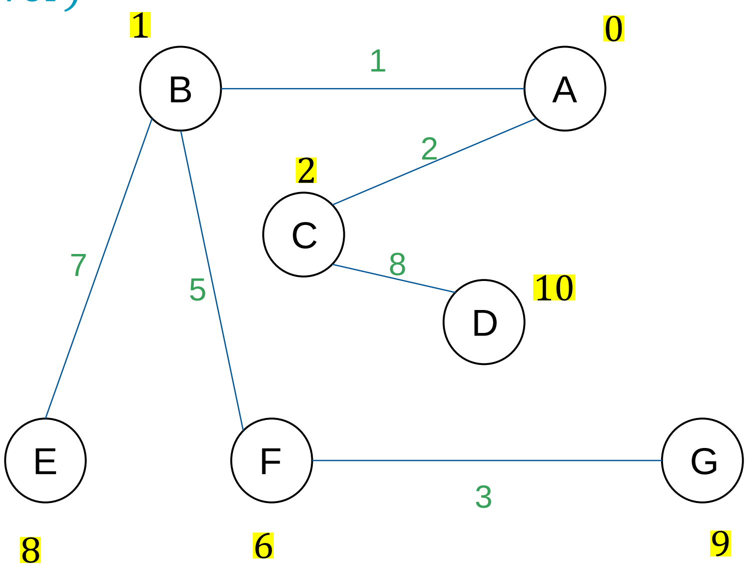


(process)



Solution

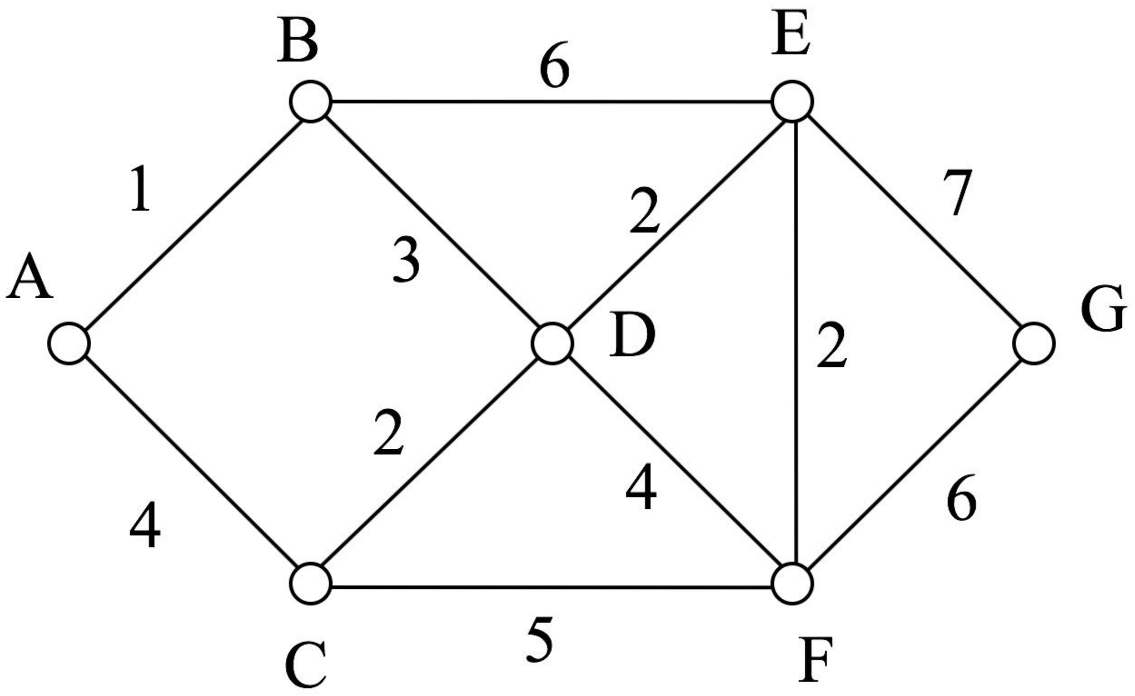
(final answer)



Vertices	Shortest path	Minimum cost
A to B	A-B	1
A to C	A-C	2
A to D	A-C-D	10
A to E	A-B-E	8
A to F	A-B-F	6
A to G	A-B-F-G	9

Practice

Find the shortest path between vertices A and G in the graph below.



Using Dijkstra's algorithm

	A	B	C	D	E	F	G
A	$\boxed{0}$	∞	∞	∞	∞	∞	∞
B		$\boxed{1}$	4	∞	∞	∞	∞
D			4	$\boxed{4}$	7	∞	∞
C			$\boxed{4}$		6	8	∞
E					$\boxed{6}$	8	∞
F						$\boxed{8}$	13
G							$\boxed{13}$

A --B---D---E---G = 1 + 3 + 2+7 = 13

Analysis

Dijkstra's algorithm is an *optimal algorithm*, meaning that it always produces the actual shortest path, not just a path that is pretty short, provided one exists. This algorithm is also *efficient*, meaning that it can be implemented in a reasonable amount of time.

Dijkstra's algorithm takes around V^2 calculations, where V is the number of vertices in the graph. A graph with 100 vertices would take around 10,000 calculations. While that would be a lot to do by hand, it is not a lot for computer to handle. It is because of this efficiency that your car's GPS unit can compute driving directions in only a few seconds.

In contrast, an *inefficient* algorithm might try to list all possible paths then compute the length of each path. Trying to list all possible paths could easily take 10^{25} calculations to compute the shortest path with only 25 vertices; that's a 1 with 25 zeros after it! To put that in perspective, the fastest computer in the world would still spend over 1000 years analysing all those paths.

SEM

- This is an evaluation of the module CELEN086
- There are 5 questions and opportunity for you to give some comments
- Scan the QR code below, and select the module CELEN086

The SEM does not require a PIN