

10 - Graph Algorithms (part 2)

[KOMS119602] & [KOMS120403]

Design and Analysis of Algorithm (2021/2022)

Dewi Sintiar

Prodi S1 Ilmu Komputer
Universitas Pendidikan Ganesha

Week ...

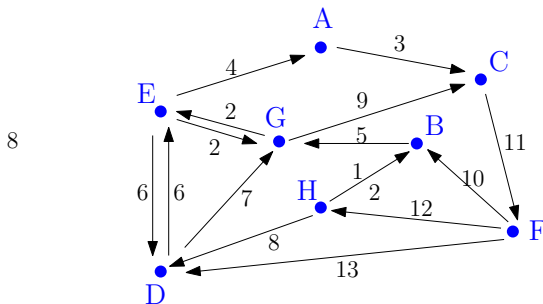
Table of contents

- Shortest path problem
 - Dijkstra algorithms

Shortest Path

Shortest path

Problem: given an edge-weighted directed graph. Find the shortest path from s to t .



Shortest path

Shortest-path tree rooted at vertex v is a spanning tree T of graph G , s.t. the path distance from root v to any other vertex u in T is the shortest path distance from v to u in G .

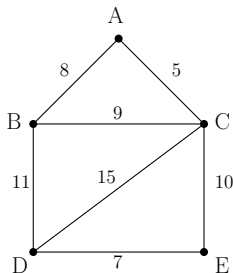


Figure: Graph G

Shortest path

Shortest-path tree rooted at vertex v is a spanning tree T of graph G , s.t. the path distance from root v to any other vertex u in T is the shortest path distance from v to u in G .

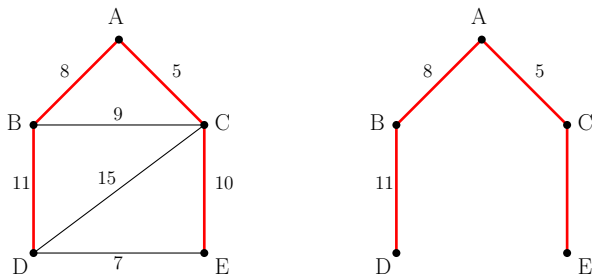


Figure: In red, the shortest path tree of G rooted at A

Shortest path

Shortest-path tree rooted at vertex v is a spanning tree T of graph G , s.t. the path distance from root v to any other vertex u in T is the shortest path distance from v to u in G .

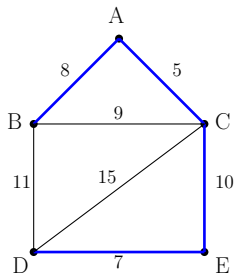
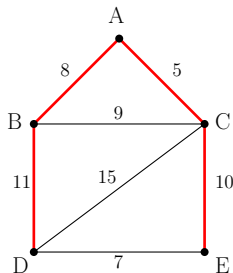


Figure: In red the shortest path tree, and in blue an MST of G

Shortest path variant

Which vertices?

- Single source: from one vertex s to every other vertex.
- Source-sink: from one vertex s to another t .
- All pairs: between all pairs of vertices.

Restrictions on edge weights?

- Nonnegative weights.
- Euclidean weights.
- Arbitrary weights.

Cycles?

- No directed cycles.
- No "negative cycles."

Applications of shortest path

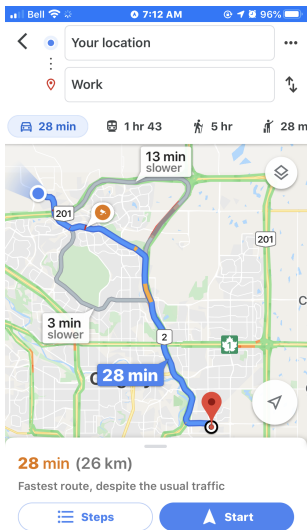


Figure: Finding shortest route in Google maps

Applications of shortest path



Figure: Car navigation system

Single-source shortest path

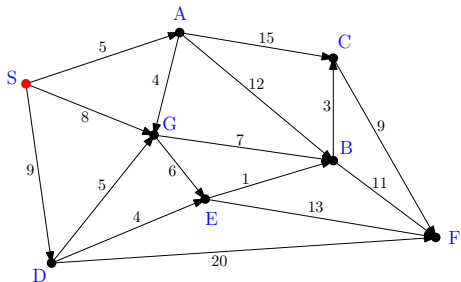
Goal: Find the shortest path from the source s to every other vertex.

Dijkstra algorithm



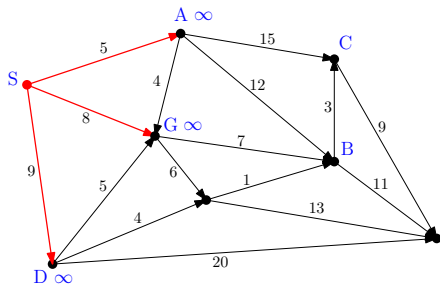
Figure: Edsger W. Dijkstra

Dijkstra algorithm



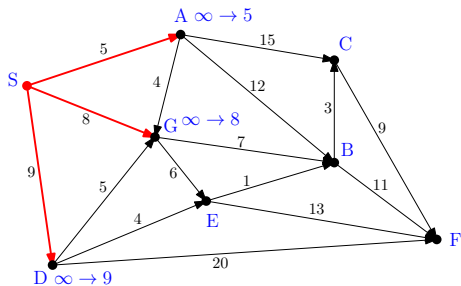
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



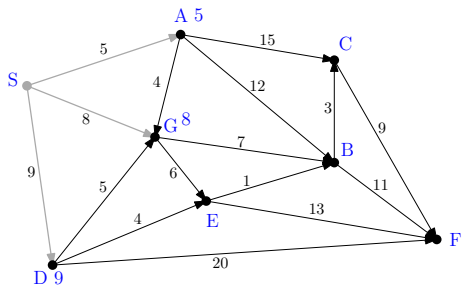
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



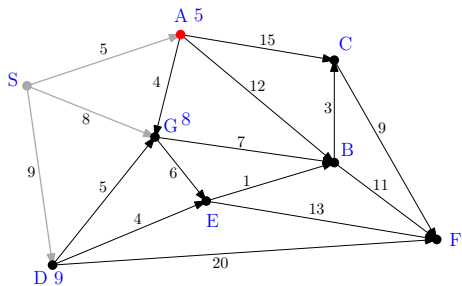
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



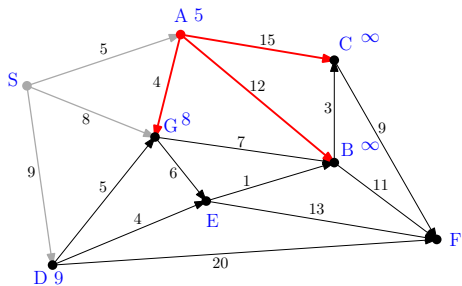
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 \rightarrow 1
2	14.0	5 \rightarrow 2
3	17.0	2 \rightarrow 3
4	9.0	0 \rightarrow 4
5	13.0	4 \rightarrow 5
6	25.0	2 \rightarrow 6
7	8.0	0 \rightarrow 7

Dijkstra algorithm



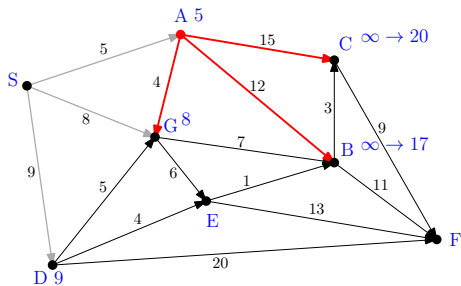
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 \rightarrow 1
2	14.0	5 \rightarrow 2
3	17.0	2 \rightarrow 3
4	9.0	0 \rightarrow 4
5	13.0	4 \rightarrow 5
6	25.0	2 \rightarrow 6
7	8.0	0 \rightarrow 7

Dijkstra algorithm



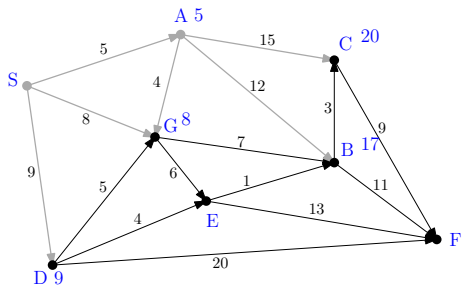
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



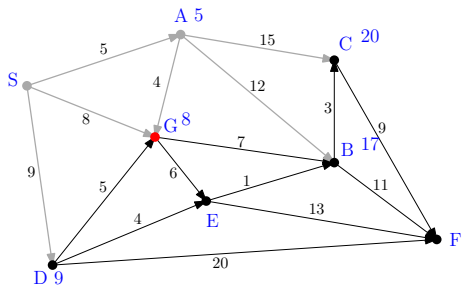
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 \rightarrow 1
2	14.0	5 \rightarrow 2
3	17.0	2 \rightarrow 3
4	9.0	0 \rightarrow 4
5	13.0	4 \rightarrow 5
6	25.0	2 \rightarrow 6
7	8.0	0 \rightarrow 7

Dijkstra algorithm



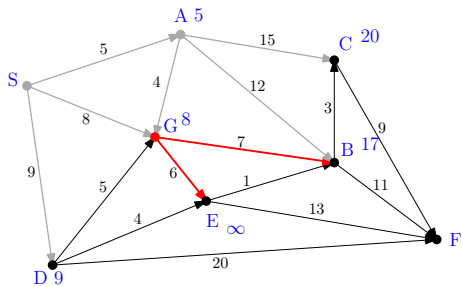
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



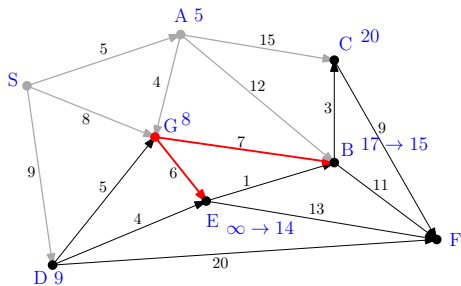
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



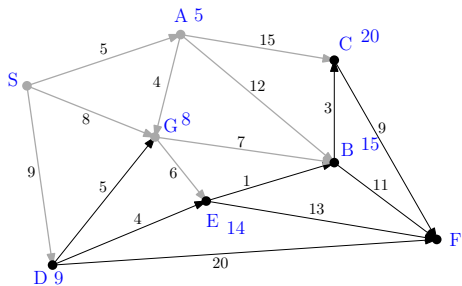
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Dijkstra algorithm



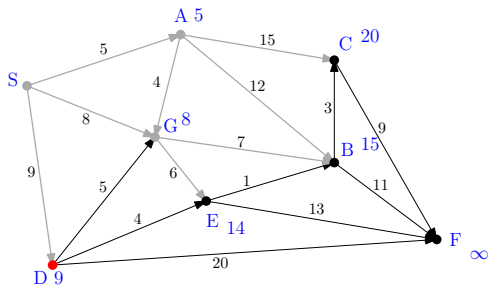
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



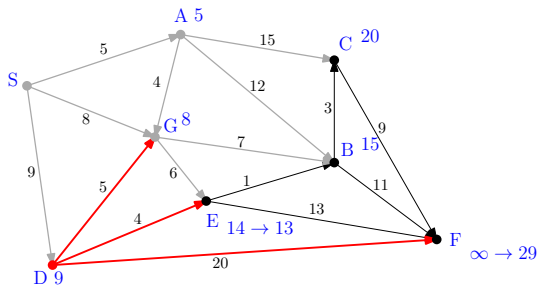
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Dijkstra algorithm



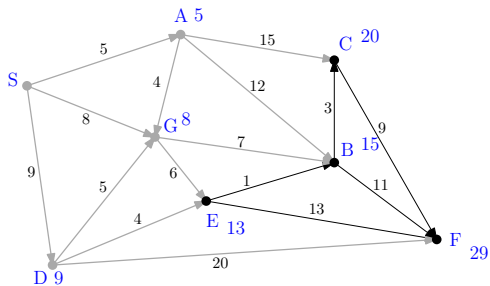
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



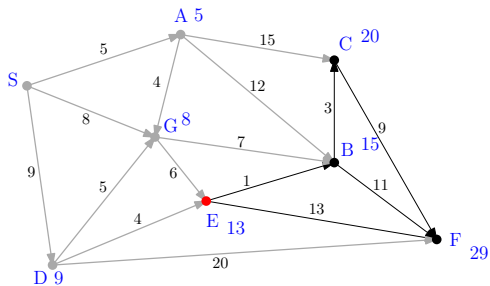
v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7

Dijkstra algorithm



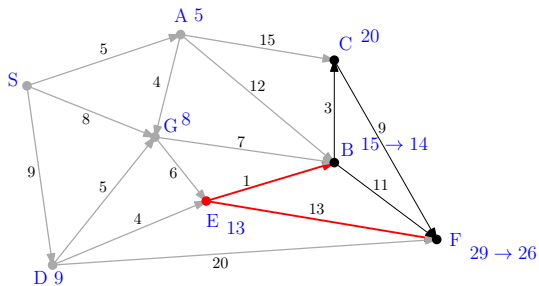
v	distTo[]	edgeTo[]
0	0	-
1	5.0	$0 \rightarrow 1$
2	14.0	$5 \rightarrow 2$
3	17.0	$2 \rightarrow 3$
4	9.0	$0 \rightarrow 4$
5	13.0	$4 \rightarrow 5$
6	25.0	$2 \rightarrow 6$
7	8.0	$0 \rightarrow 7$

Dijkstra algorithm



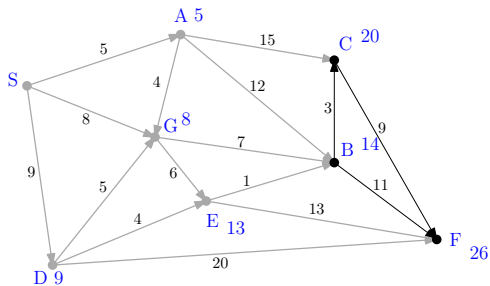
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



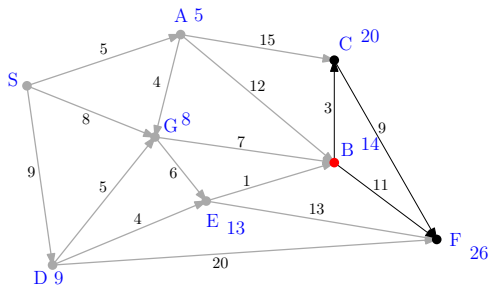
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



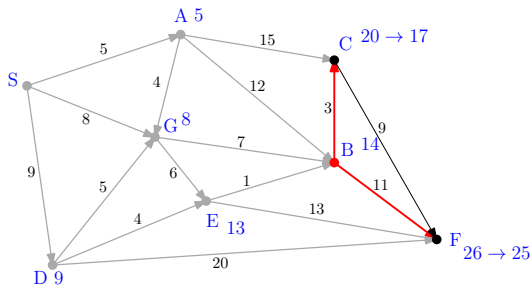
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



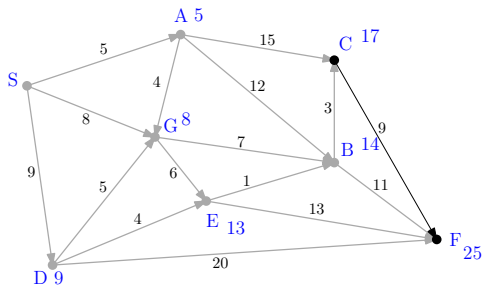
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



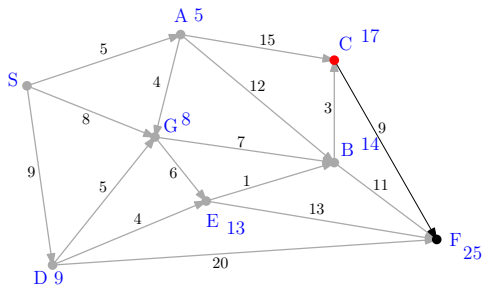
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



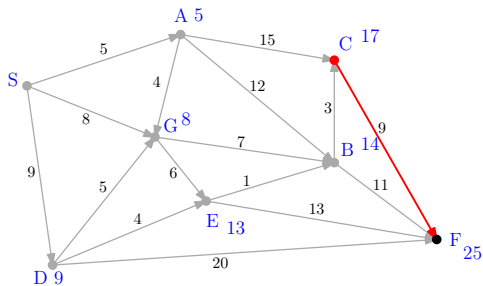
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



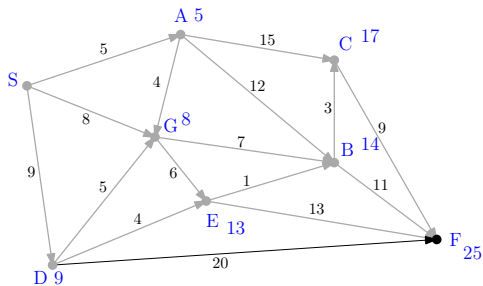
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



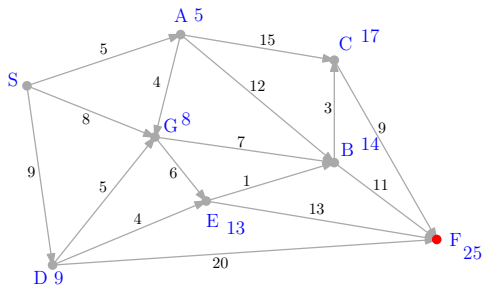
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



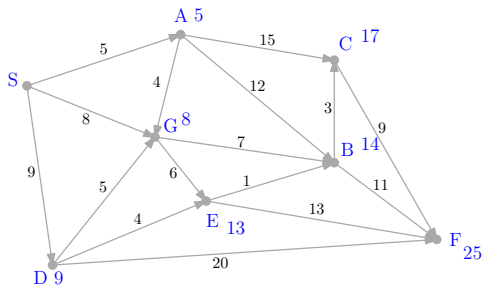
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



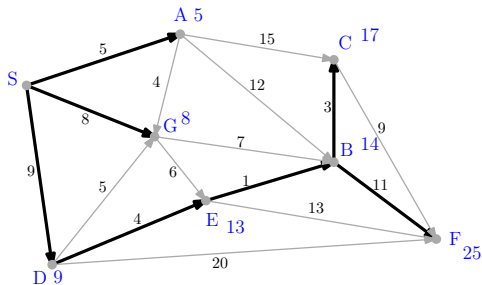
v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



v	distTo[]	edgeTo[]
0	0	
1	5.0	
2	14.0	
3	17.0	
4	9.0	
5	13.0	
6	25.0	
7	8.0	

Dijkstra algorithm



v	distTo[]	edgeTo[]
0	0	-
1	5.0	0 → 1
2	14.0	5 → 2
3	17.0	2 → 3
4	9.0	0 → 4
5	13.0	4 → 5
6	25.0	2 → 6
7	8.0	0 → 7