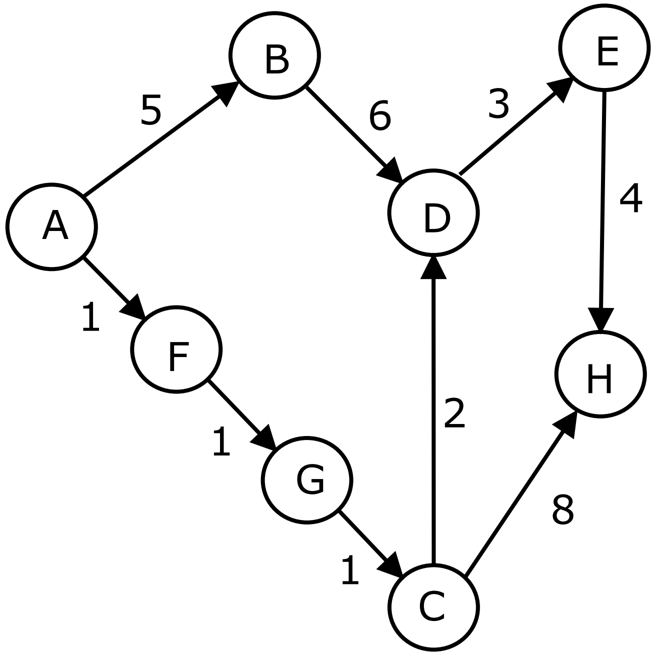
Directed Weighted Graphs

Prepared by Mahdi Ghamkhari

Directed Weighted Graphs

In a directed weighted graph each edge has a weight and a direction

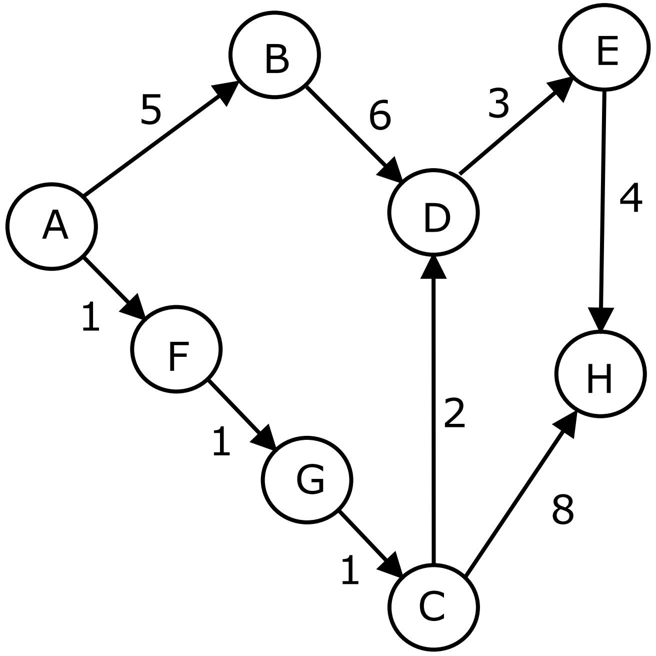
For instance, A-B is an edge from vertex A to vertex B and has a weight of 5



Implementation: Adjacency Matrix

```
Adjacency Matrix:


      A     B     C     D     E     F     G     H
A    -1     5    -1    -1    -1     1    -1    -1
B    -1    -1    -1     6    -1    -1    -1    -1
C    -1    -1    -1     2    -1    -1    -1     8
D    -1    -1    -1    -1     3    -1    -1    -1
E    -1    -1    -1    -1    -1    -1    -1     4
F    -1    -1    -1    -1    -1    -1     1    -1
G    -1    -1     1    -1    -1    -1    -1    -1
H    -1    -1    -1    -1    -1    -1    -1    -1
```



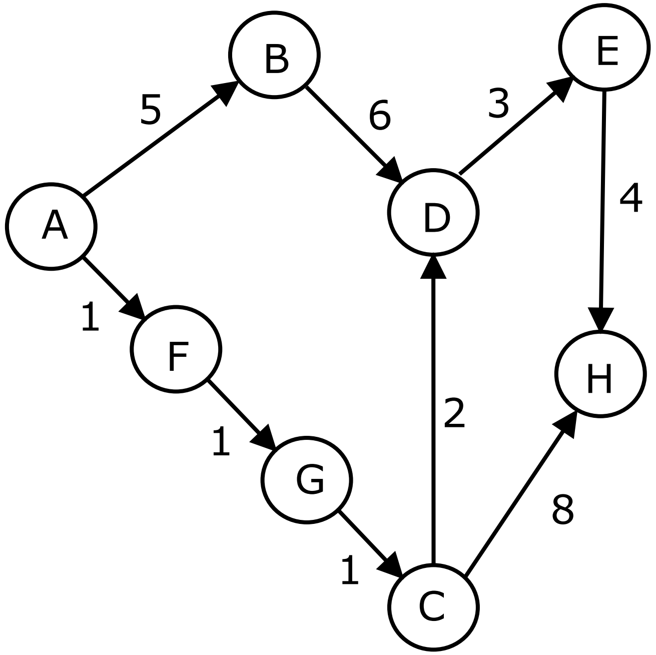
Paths

From A to H there are different paths:

A-B-D-E-H

A-F-G-C-H

A-F-G-C-D-E-H



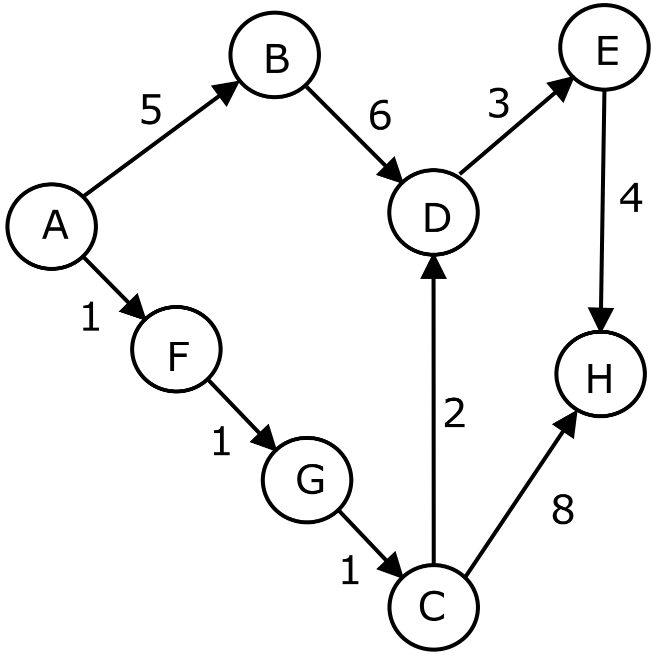
Length of Paths

From A to H there are different paths:

A-B-D-E-H: 5+6+3+4 = 18

A-F-G-C-H: 1+1+1+8 = 11

A-F-G-C-D-E-H: 1+1+1+2+3+4 = 12



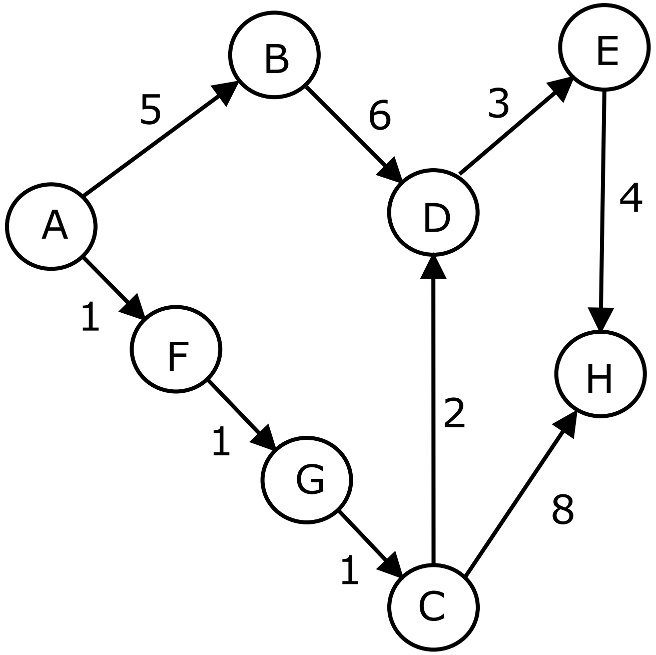
Distance from Vertex A to Vertex H

Distance is the length of the shortest path

A-B-D-E-H: 5+6+3+4 = 18

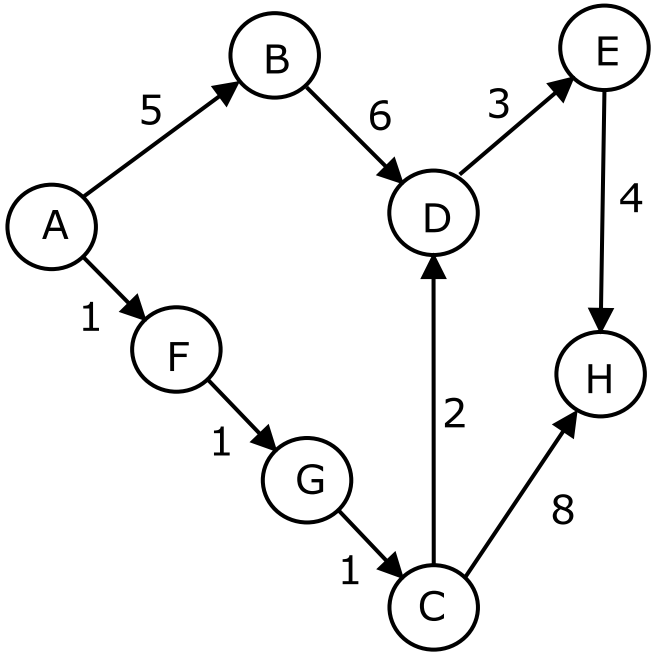
A-F-G-C-H: 1+1+1+8 = 11

A-F-G-C-D-E-H: 1+1+1+2+3+4 = 12



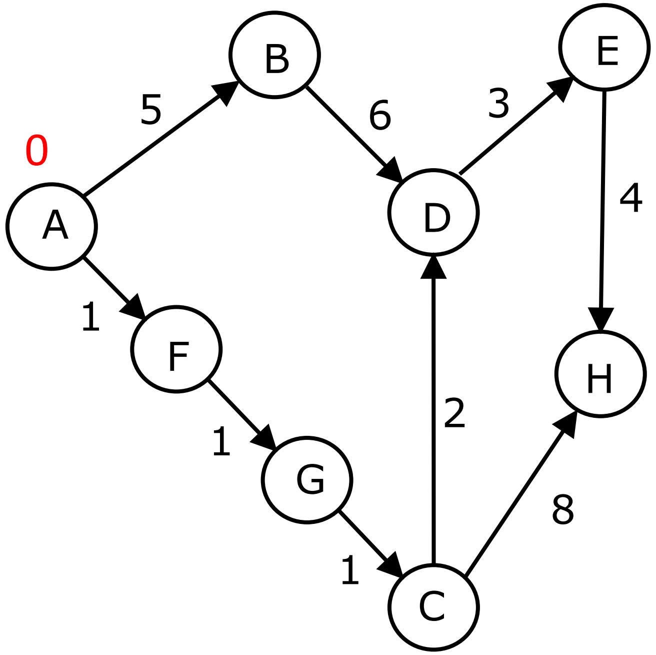
Finding Distances from a Source Vertex

- What is the distance from A to B?
- What is the distance from A to C?
- What is the distance from A to F?
- What is the distance from A to G?
- What is the distance from A to D?
- What is the distance from A to E?
- What is the distance from A to H?



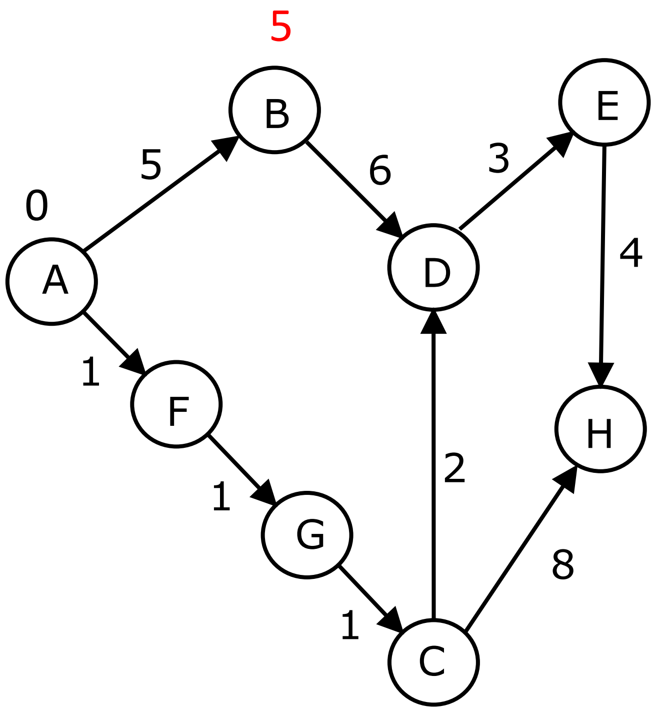
Finding Distances from a Source Vertex

- Distance of A from A is 0



Finding Distances from a Source Vertex

- From A to B there is an edge with a weight of 5
- Distance from A to B is updated to 5
- B is queued

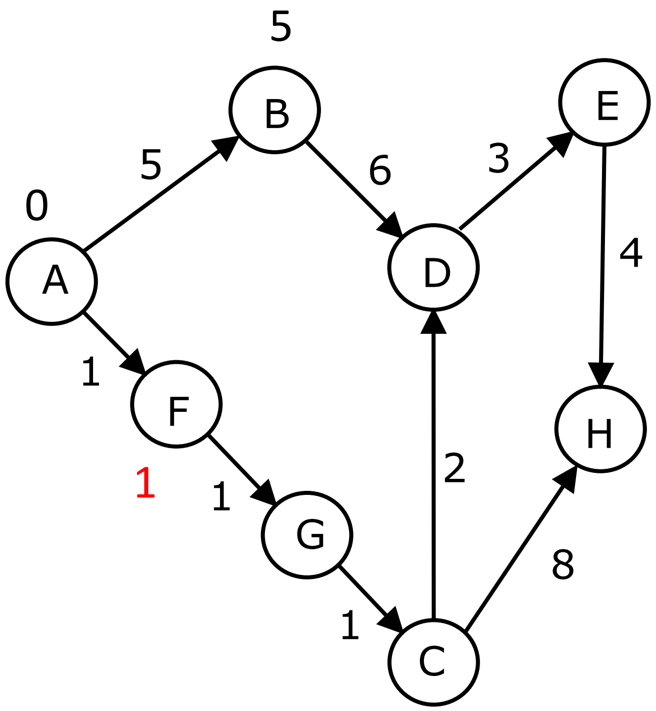


Queue

B

Finding Distances from a Source Vertex

- From A to F there is an edge with a weight of 1
- Distance from A to F is updated to 1
- F is queued

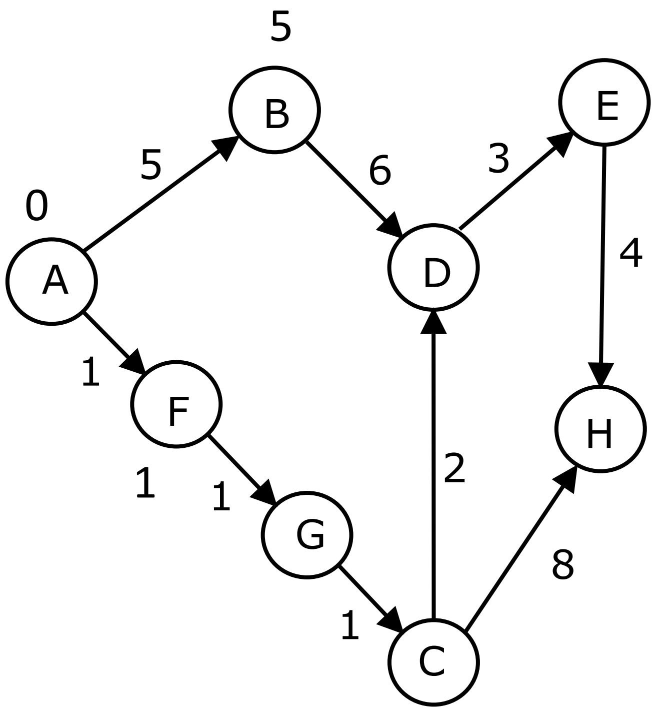


Queue

B F

Finding Distances from a Source Vertex

- A has no other neighbors
- We take a vertex from the Queue: F
- Since F has the lowest distance from A

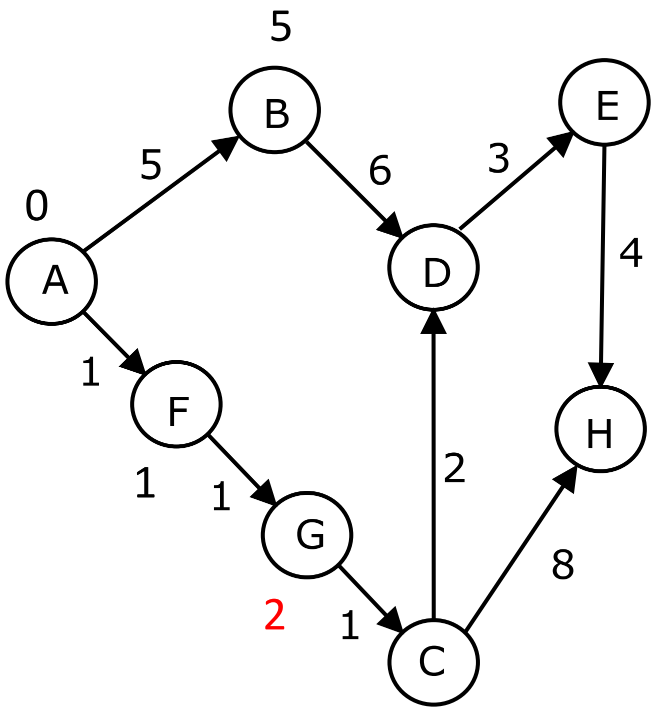


Queue

B F

Finding Distances from a Source Vertex

- From F to G there in an edge with a weight of 1
- Distance from A to F is updated to 1+1=2
- G is queued

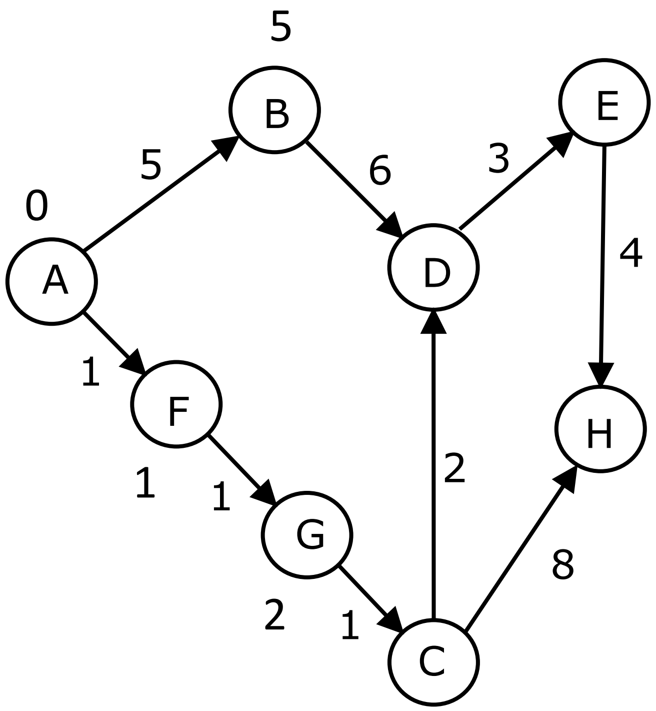


Queue

B G

Finding Distances from a Source Vertex

- F has no other neighbors
- We take a vertex from the Queue: G
- Since G has the lowest distance from A

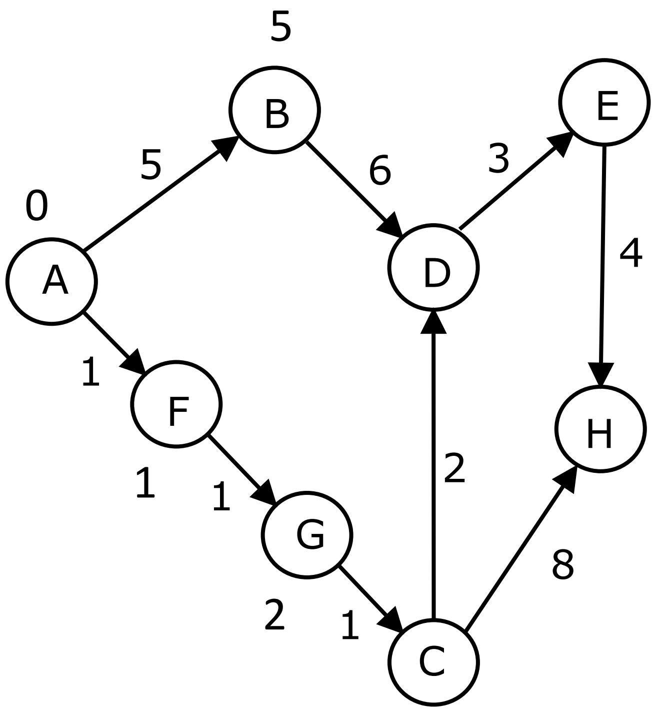


Queue

B G

Finding Distances from a Source Vertex

- From G to C there in an edge with a weight of 1
- Distance from A to F is updated to 2+1=3
- C is queued

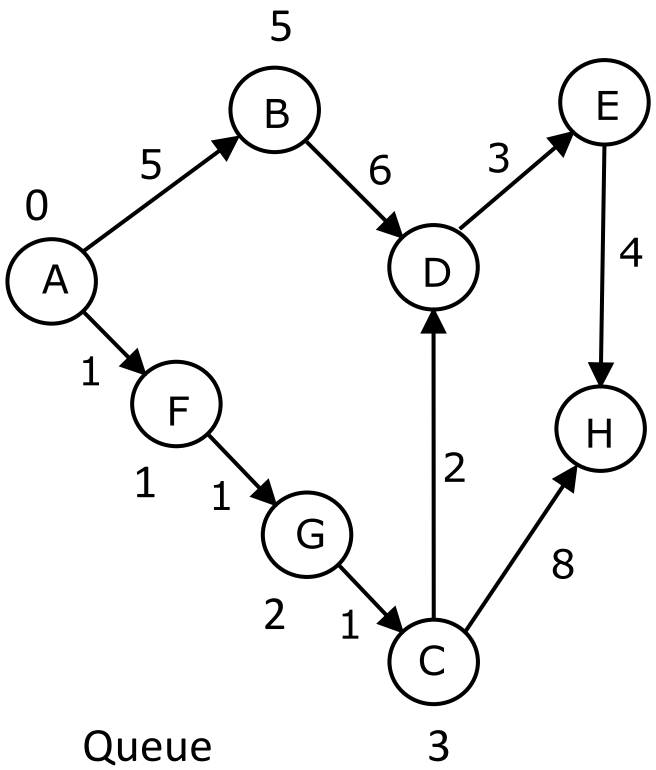


Queue 3

B C

Finding Distances from a Source Vertex

- G has no other neighbors
- We take a vertex from the Queue: C
- Since C has the lowest distance from A

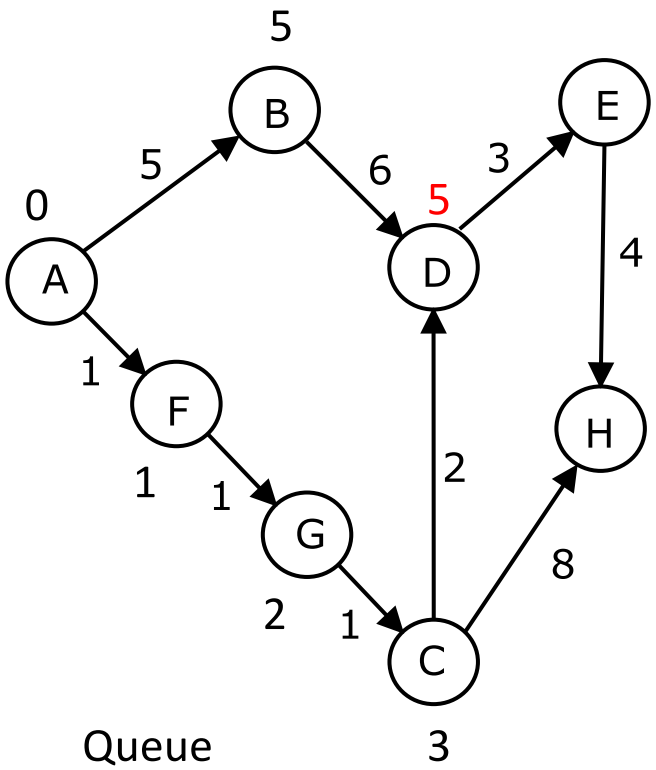


Queue

B C

Finding Distances from a Source Vertex

- From C to D there in an edge with a weight of 2
- Distance from A to D is updated to 3+2=5
- D is queued

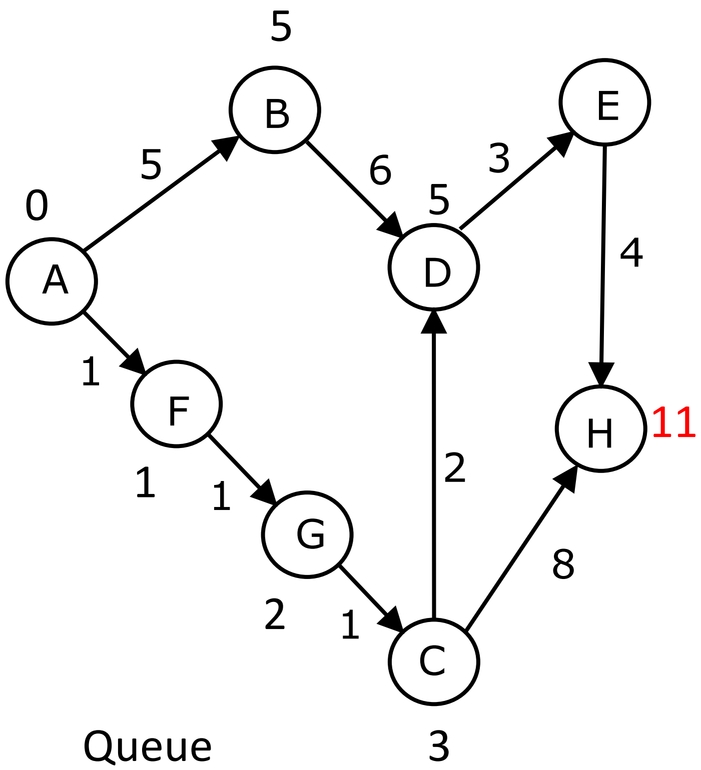


Queue

B D

Finding Distances from a Source Vertex

- From C to H there in an edge with a weight of 8
- Distance from A to H is updated to 3+8=11
- H is queued

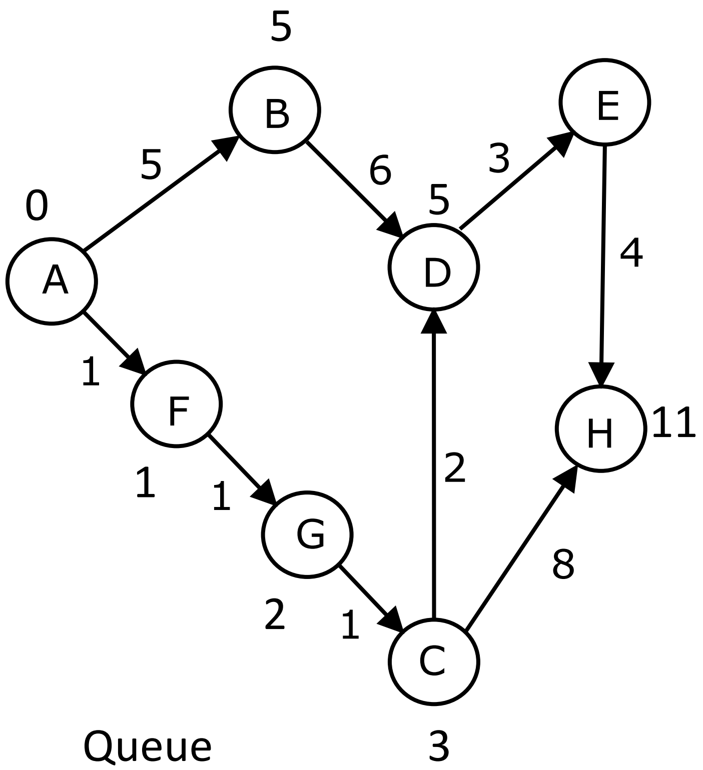


Queue

B D H

Finding Distances from a Source Vertex

- C has no other neighbors
- We take a vertex from the Queue: B
- Since B has the lowest distance from A
- Between B and D we choose B since it was placed in the queue first

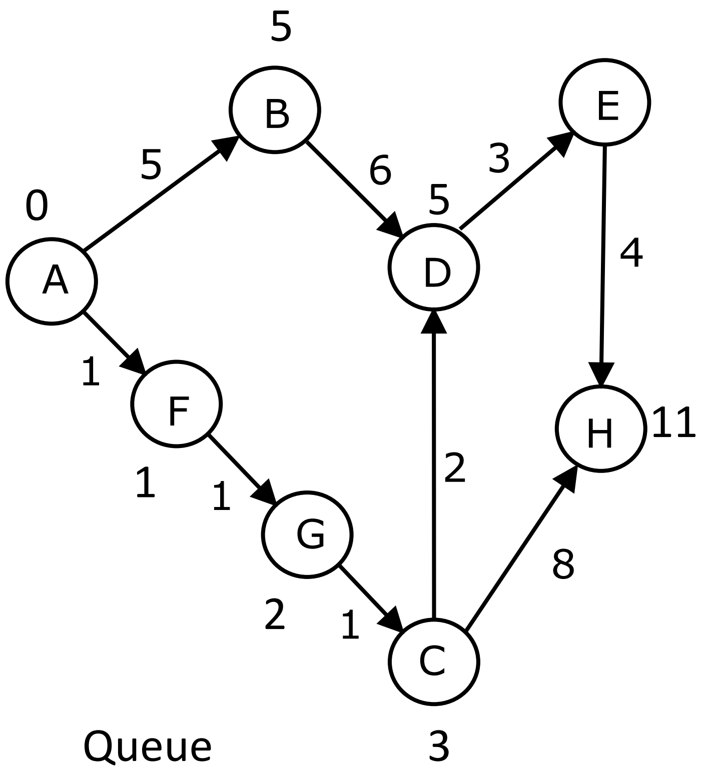


Queue

B D H

Finding Distances from a Source Vertex

- From B to D there in an edge with a weight of 6
- Distance from A to D is not updated to 5+6=11
- D is not queued, since its distance was not updated

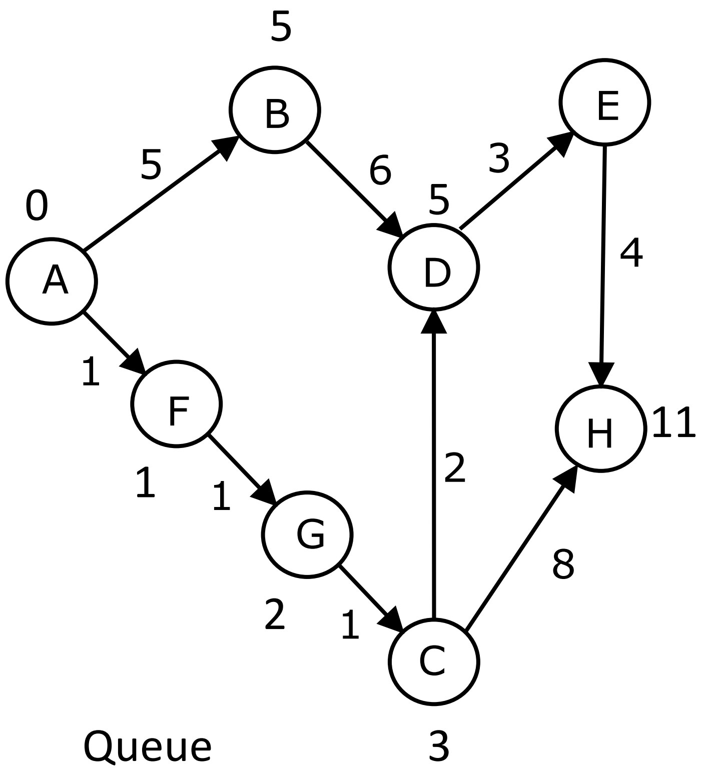


Queue

D H

Finding Distances from a Source Vertex

- B has no other neighbors
- We take a vertex from the Queue: D
- Since D has the lowest distance from A

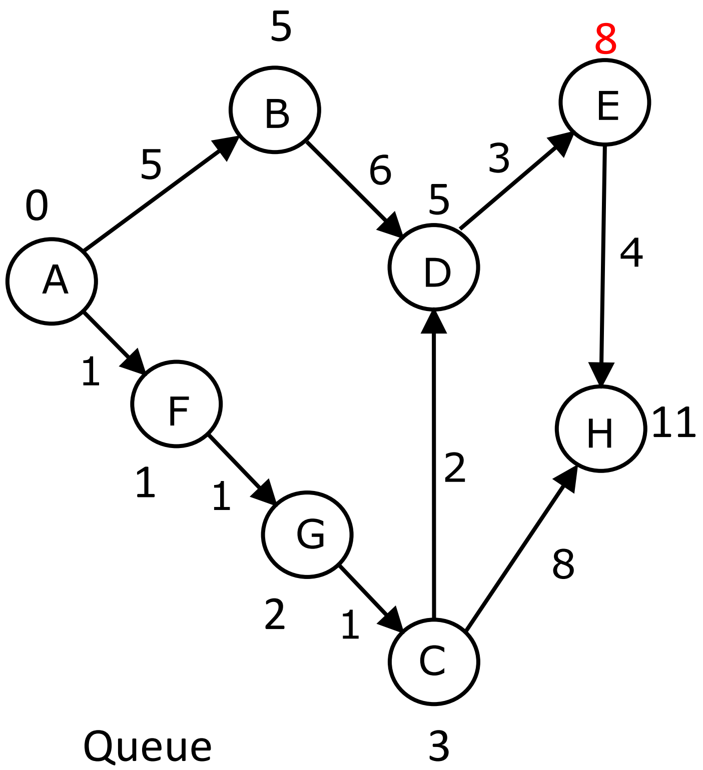


Queue

D H

Finding Distances from a Source Vertex

- From D to E there is an edge with a weight of 3
- Distance from A to E is updated to 5+3=8
- E is queued

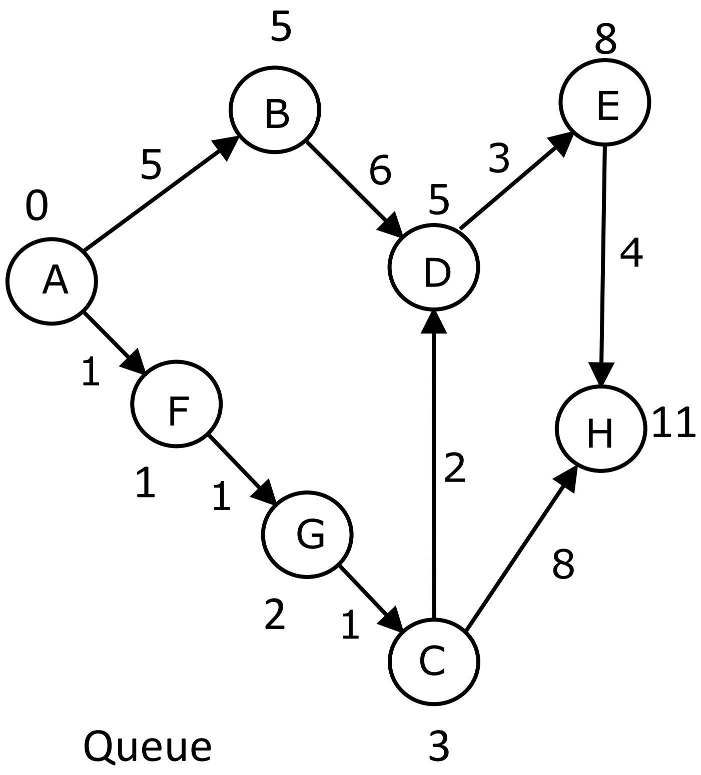


Queue

H E

Finding Distances from a Source Vertex

- D has no other neighbors
- We take a vertex from the Queue: E
- Since E has the lowest distance from A

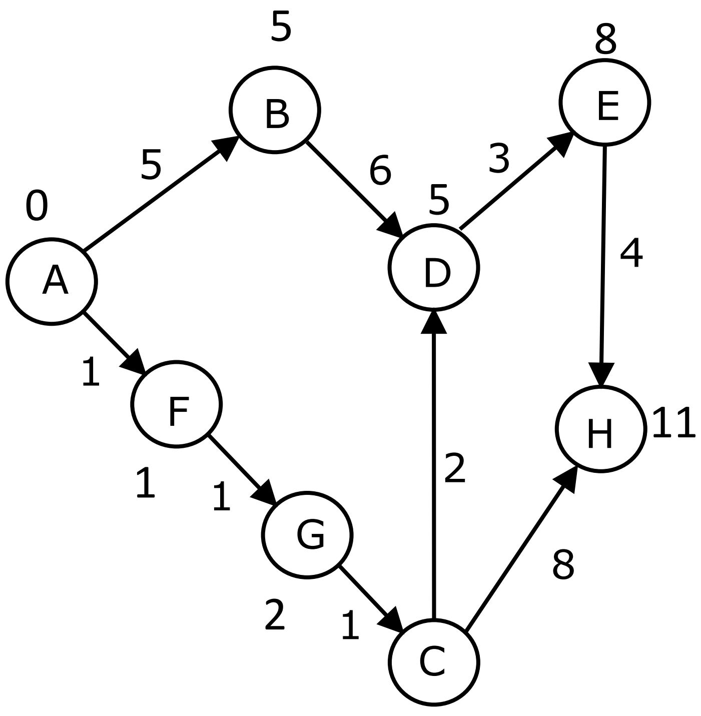


Queue

H E

Finding Distances from a Source Vertex

- From E to H there in an edge with a weight of 4
- Distance from A to H is not updated to 8+4=12
- H is not queued, since its distance was not updated



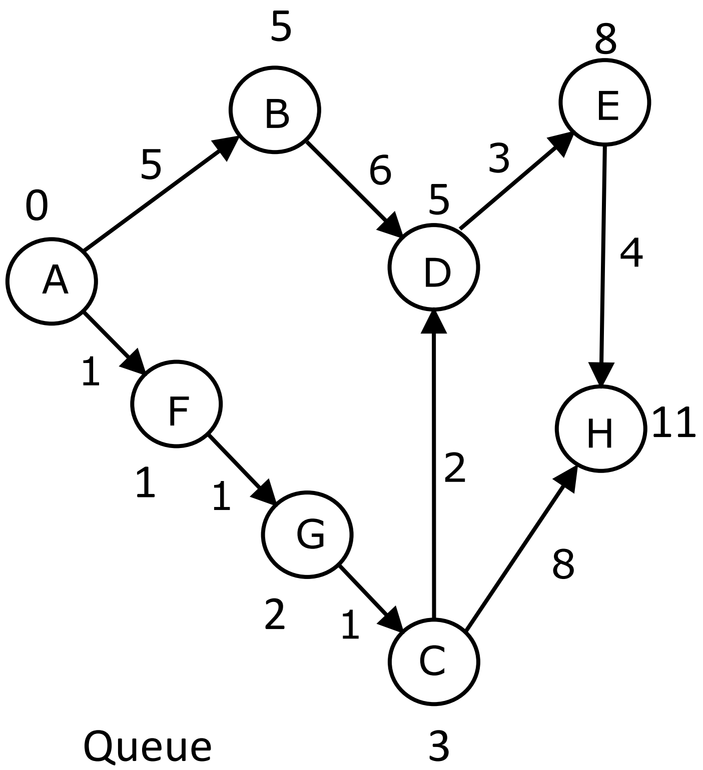
Queue

3

H

Finding Distances from a Source Vertex

- E has no other neighbors
- We take a vertex from the Queue: H
- Since H has the lowest distance from A

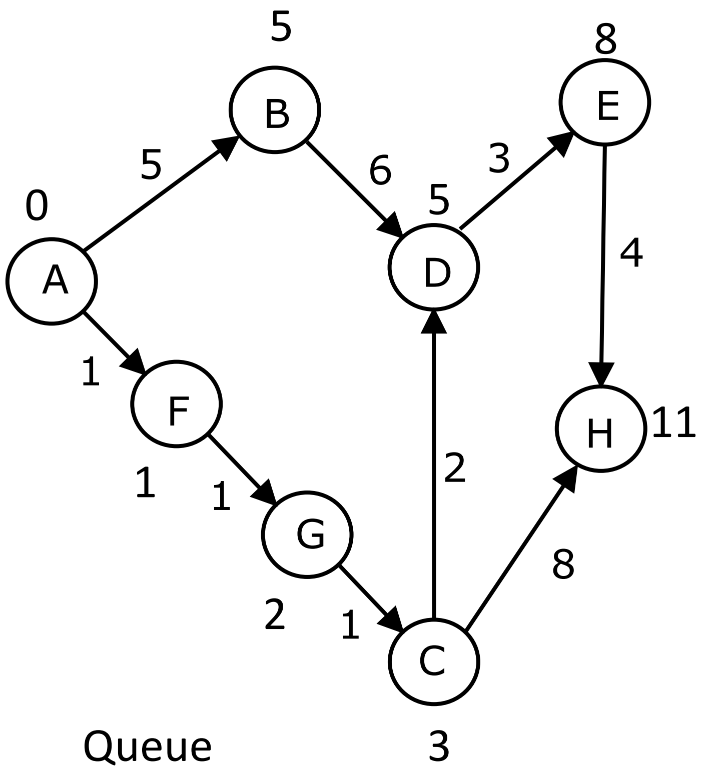


Queue

H

Finding Distances from a Source Vertex

- H has no other neighbors
- We take a vertex from the Queue
- But the Queue is empty
- Algorithms is finished



Queue

- The Queue in this algorithm is a priority queue, because vertexes are deQueued from the Queue according to their distances from A
- A priority queue can be based on arrays or trees
- A priority queue based on trees has a better time complexity
- A priority queue based on trees is a heap
- We use a heap for finding distances in a directed weighted graph