Graphs

Single Source Shortest Paths for DAGs

Stefano Quer
Dipartimento di Automatica e Informatica
Politecnico di Torino

Shortest paths on weighted DAGs

- For a DAG the SSSPs problem can be solved with a simplified algorithm
- Shortest paths are always well defined even if there are negative-weight edges
  - This is because, obviously, negative-weight cycles cannot exist in a DAG

Shortest paths on weighted DAGs

- As there are no cycles it is enough to
  - Topologically sort the DAG
    - Impose a linear order on the vertices

Perform a DFS computing end-processing times
Order vertices using the end-processing times

  - Relax all vertices following the sorted order given by the topological sort
    - In other words, it suffices to make just one pass over the vertices in the topological sorted order
    - As we process a vertex, we relax each edge that leaves the vertex

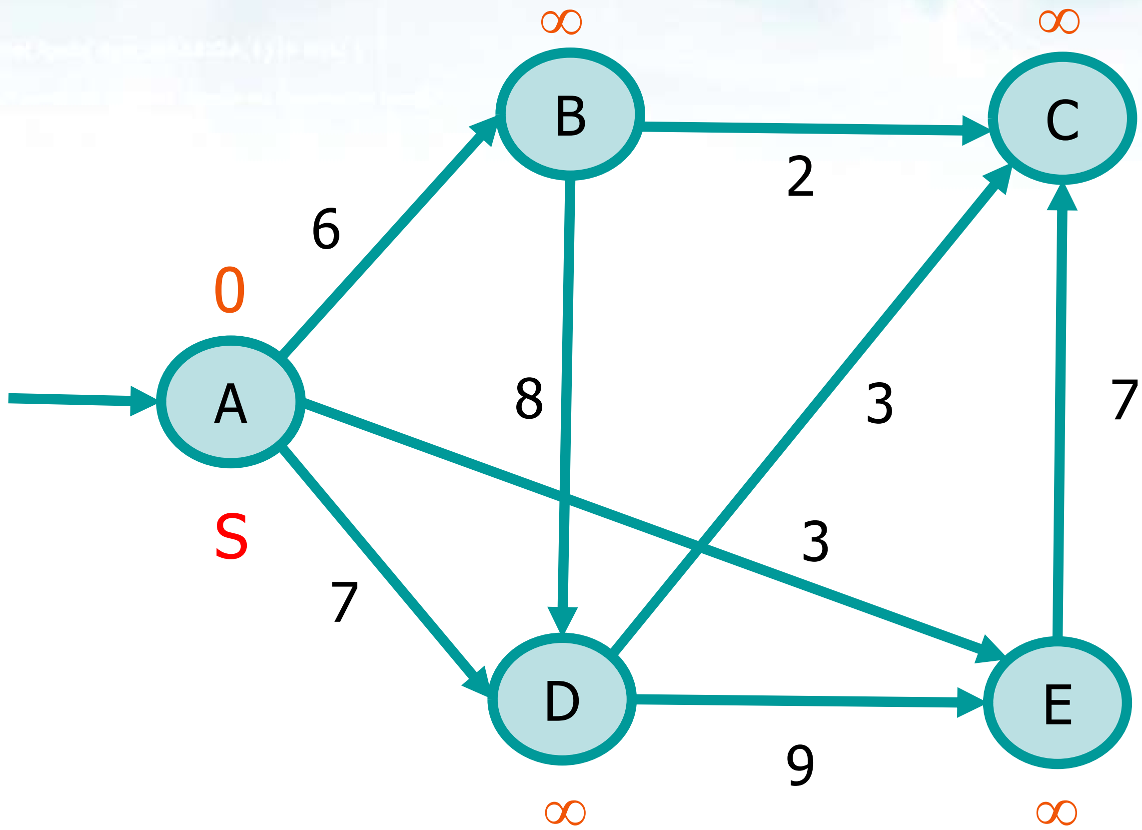
SSSP for DAGs

Pseudo-code

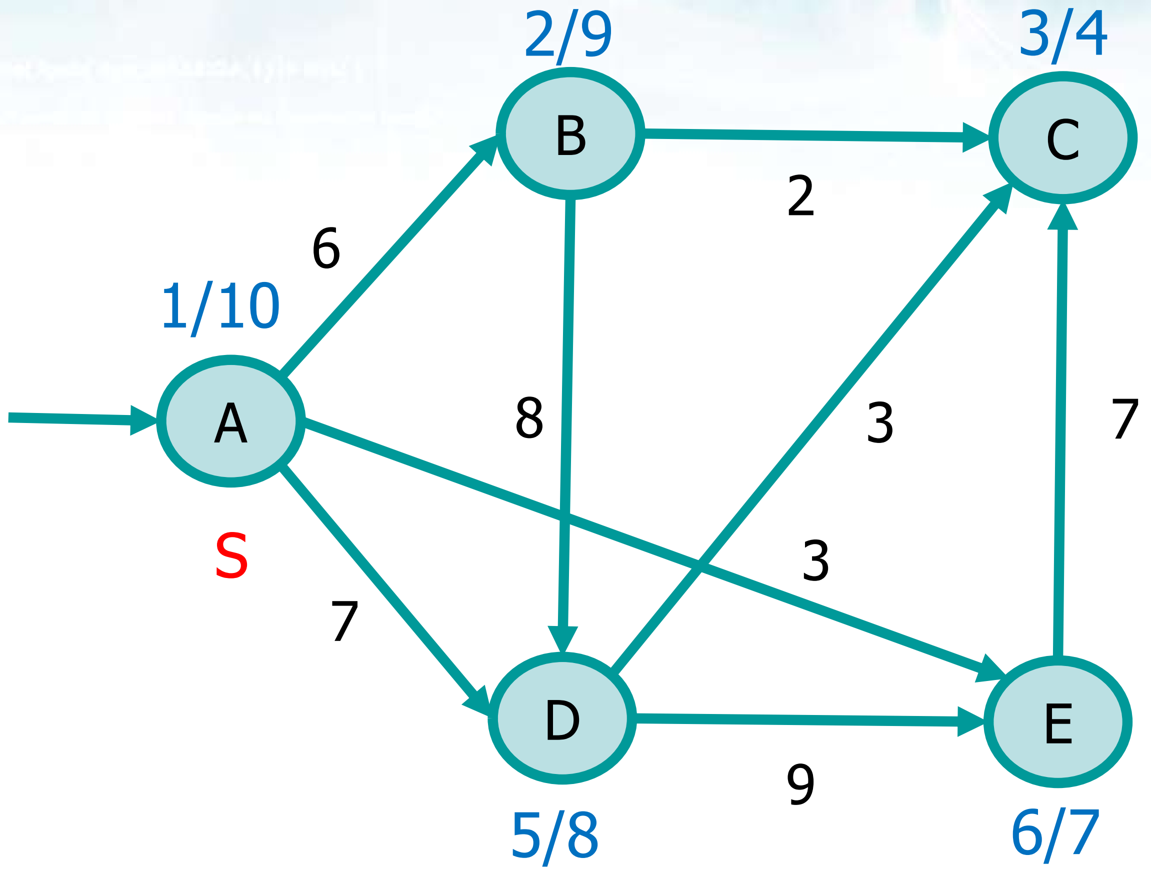
```
sssp_for_DAGs (G, w, s)
  topological sort the vertices of G
  initialize_single_source (G, s)

  for each vertex u ∈ V
    for each vertex v ∈ adjacency list of u
      relax (u, v, w)
```

Taken in topologically sorted order

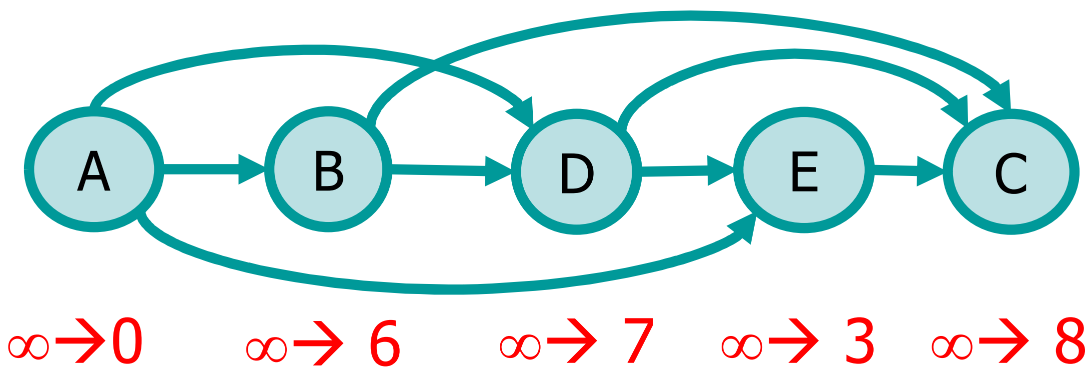
Example



Solution



Relaxation order
(A, B)
(A, D)
(A, E)
(B, D)
(B, C)
(D, E)
(D, C)
(E, C)



Complexity

Pseudo-code

```
sssp_for_DAGs (G, w, s)
  topological sort the vertices of G
  initialize_single_source (G, s)

  for each vertex u ∈ V
    for each vertex v ∈ adjacency list of u
      relax (u, v, w)
```

$\Theta(|V|+|E|)$

$\Theta(|V|)$

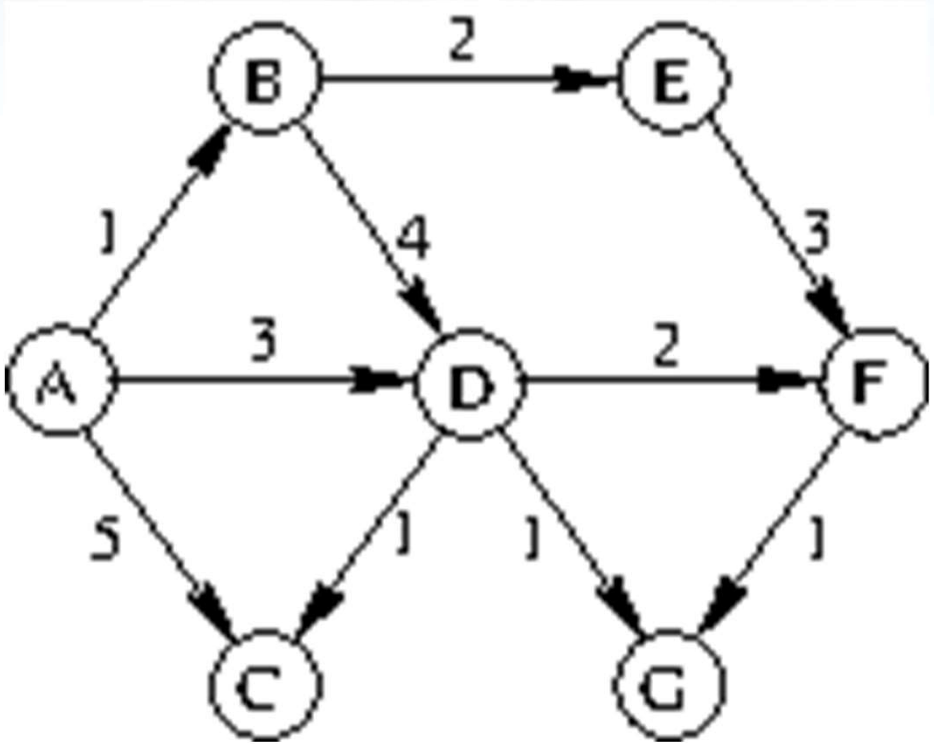
Executed E times alltogheter

$\Theta(1) \rightarrow \Theta(|E|)$

Taken in topological sorted order

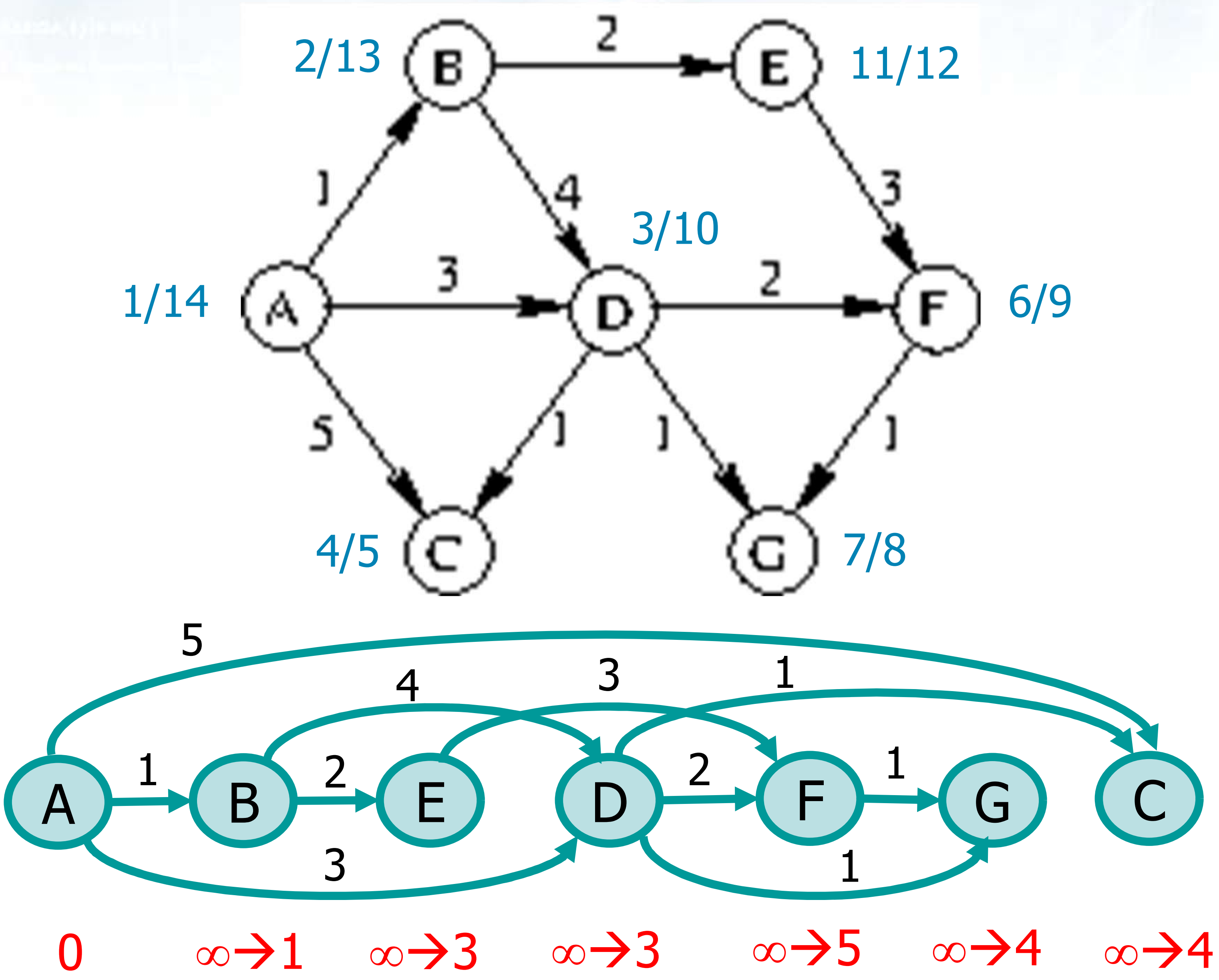
Overall running time complexity
$T(n) = \Theta(|V| + |E|)$

Exercise

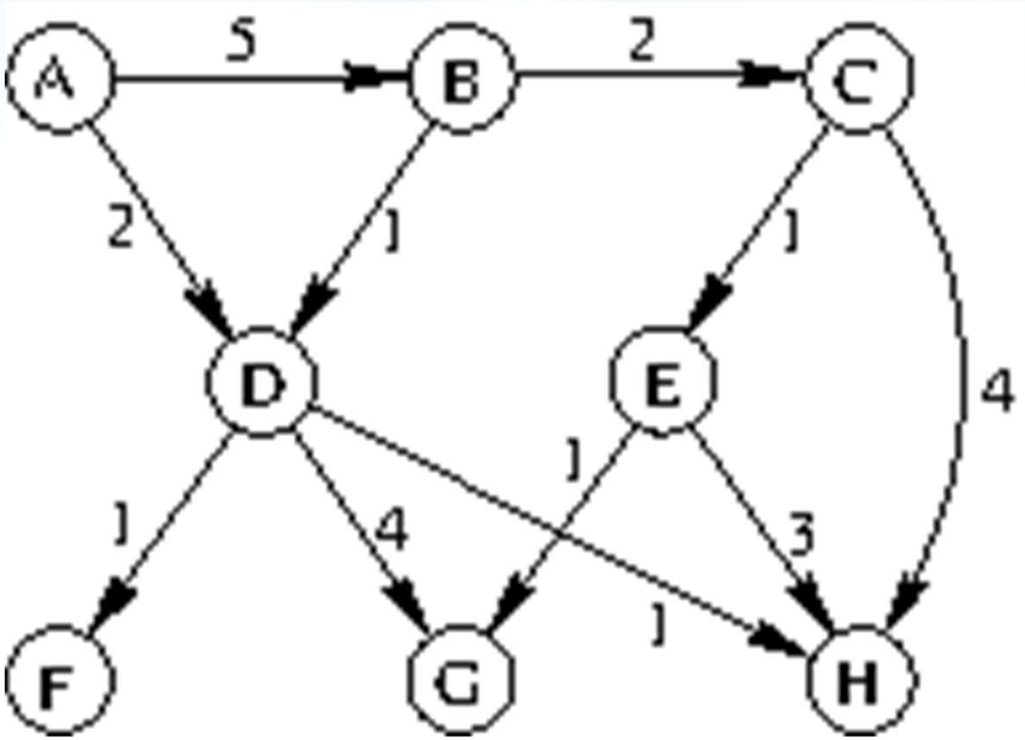


- Given the following graph find all shortest paths starting from vertex A

Solution

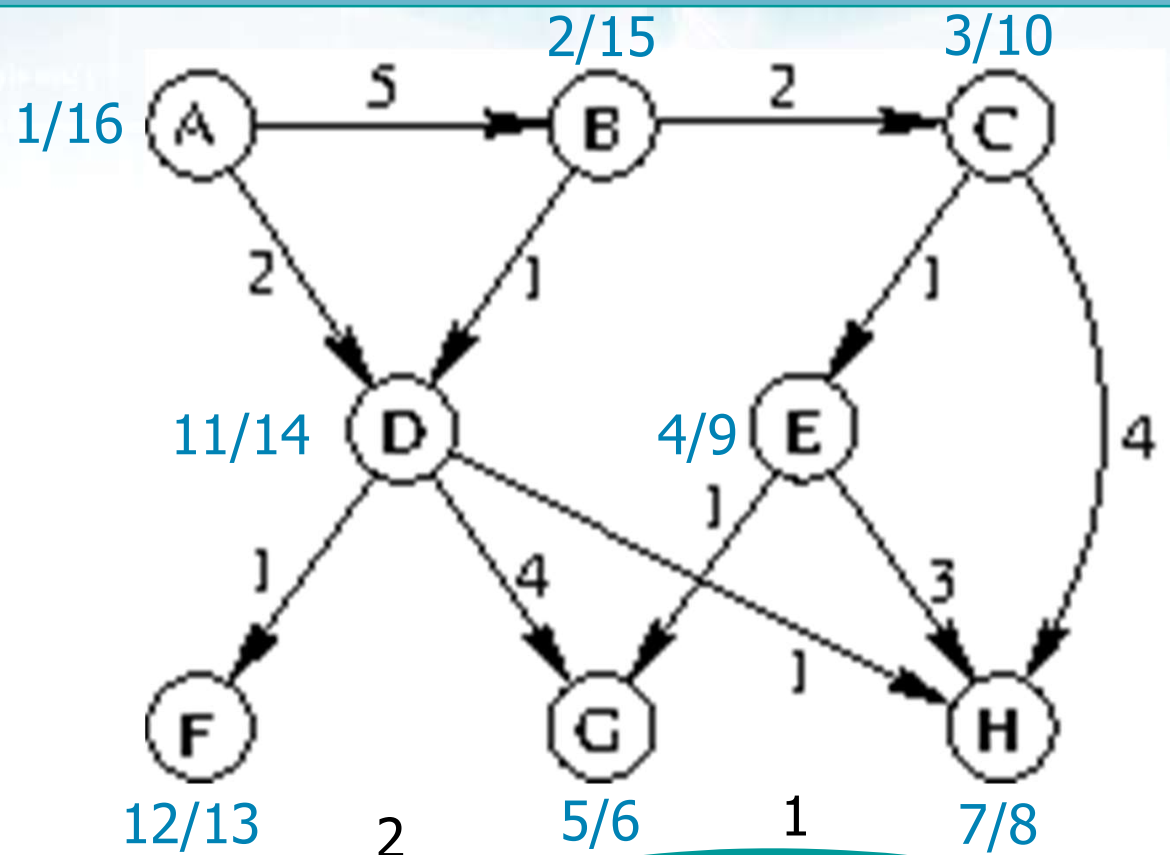

2/13
B
2
E
11/12
1
4
3
3/10
1/14
A
3
D
2
F
6/9
5
1
1
1
4/5
C
G
7/8
5
4
3
1
1
2
2
1
A
B
E
D
F
G
C
3
1
0
∞→1
∞→3
∞→3
∞→5
∞→4
∞→4

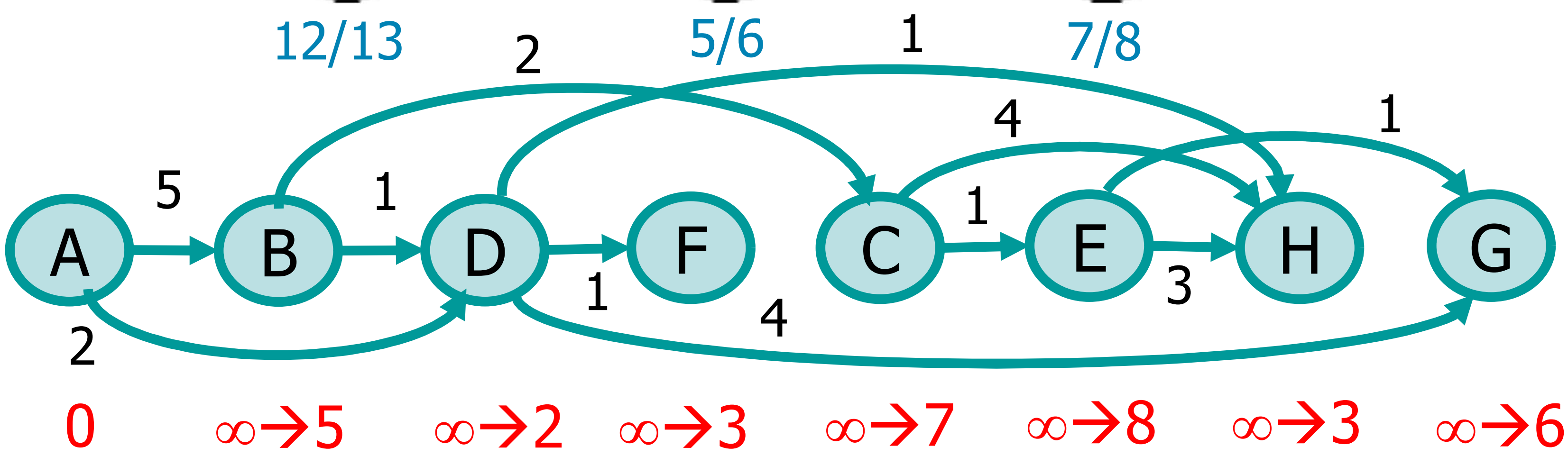
Exercise



- Given the following graph find all shortest paths starting from vertex A

Solution


2/15
3/10
1/16
A
5
B
2
C
2
1
1
11/14
D
4/9
E
4
1
1
4
3
1
F
G
H
12/13
5/6
7/8

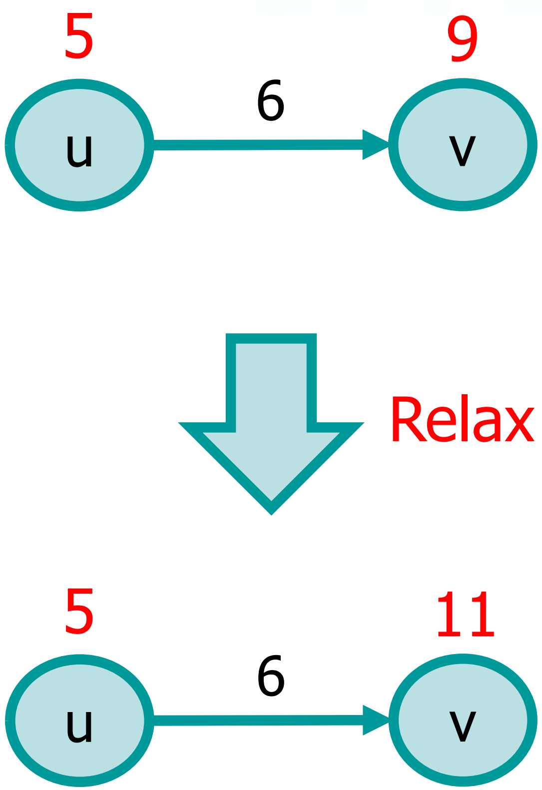

2
1
4
1
5
1
1
A
B
D
F
C
E
H
G
1
3
2
4
0
∞→5
∞→2
∞→3
∞→7
∞→8
∞→3
∞→6

Longest path on weighted DAG

- Problem intractable on generic weighted graph
- As on a DAG there are no cycles, the problem become computationally feasible
  - Topologically sort the DAG
  - For all ordered vertices
    - Apply the "inverse" relaxation rule starting from that vertex

```
inverse_relax (u, v, w) {
  if (v.dist < u.dist + w(v,u)) {
    v.dist = u.dist + w(v,u)
    v.pred = u
  }
}
```

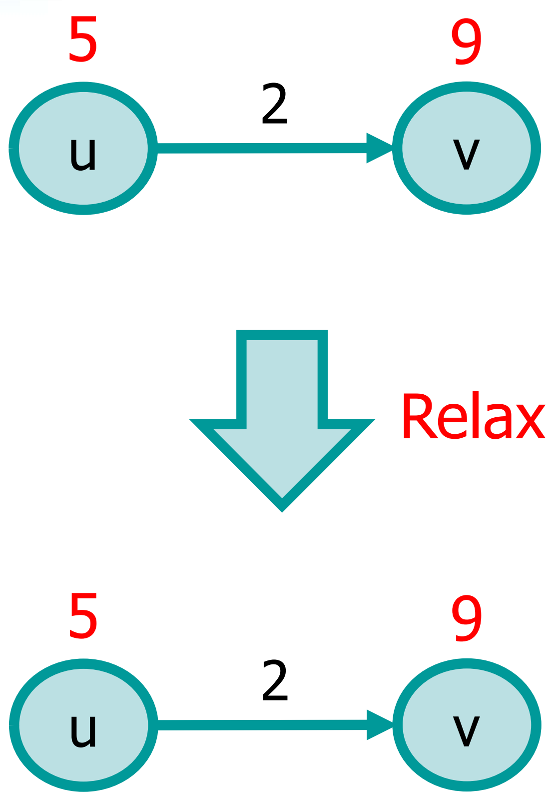
Example



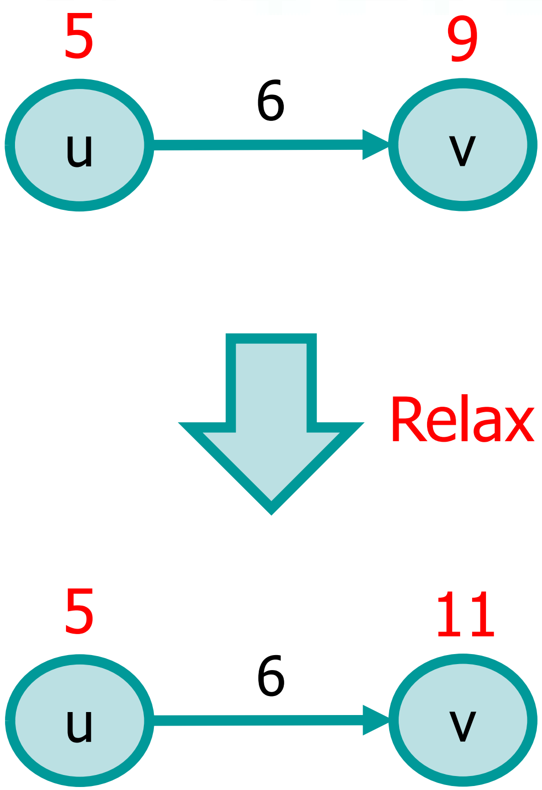

```
v.dist = 9
u.dist = 5
w(u,v) = 2
v.dist <
  u.dist + w(u,v)
```

Longest path from s to v = longtest path from s to u + edge (u,v)

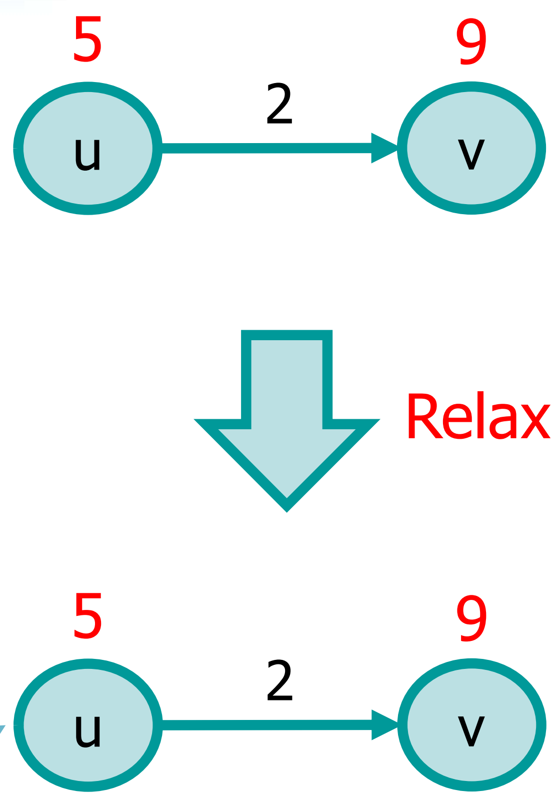
```
v.dist =
     = u.dist + w(u,v)
     = 5 + 2 = 7
v.pred = u
```



Example



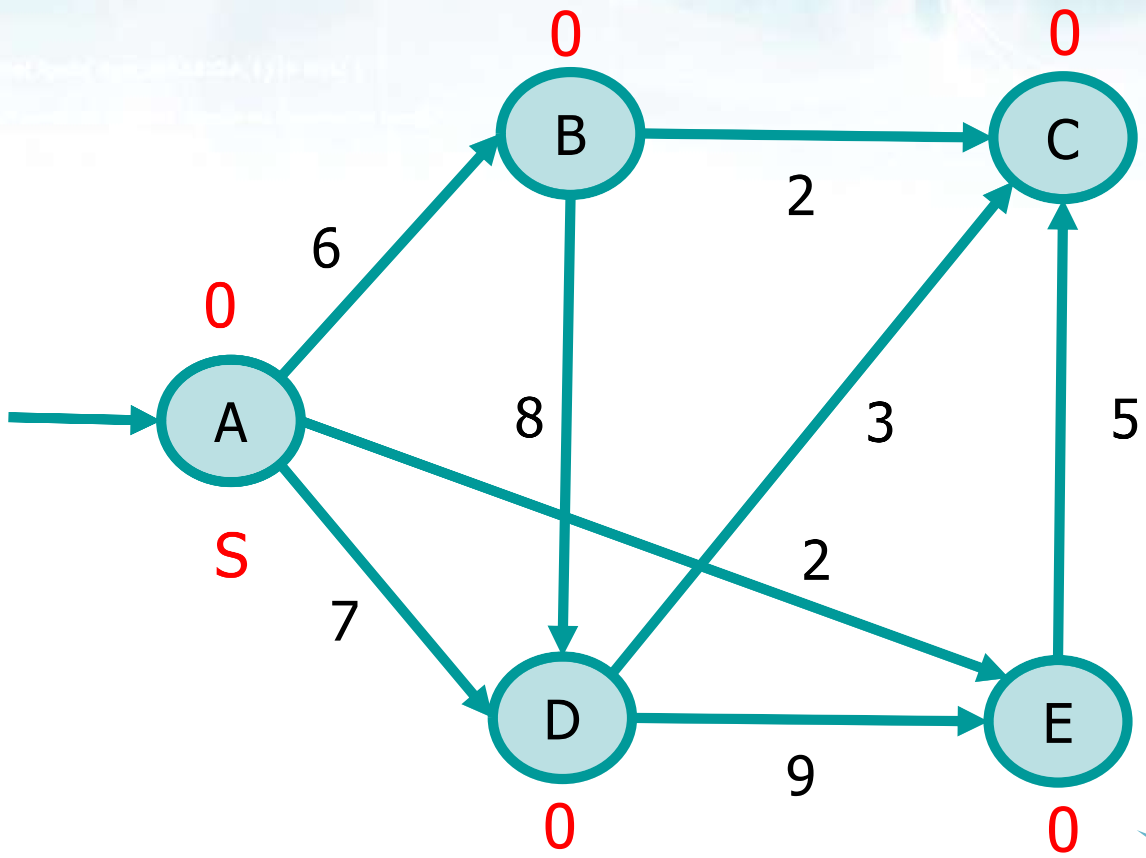

```
v.dist = 9
u.dist = 5
w(u,v) = 2
v.dist >
  u.dist + w(u,v)
```



Relaxation has no effect

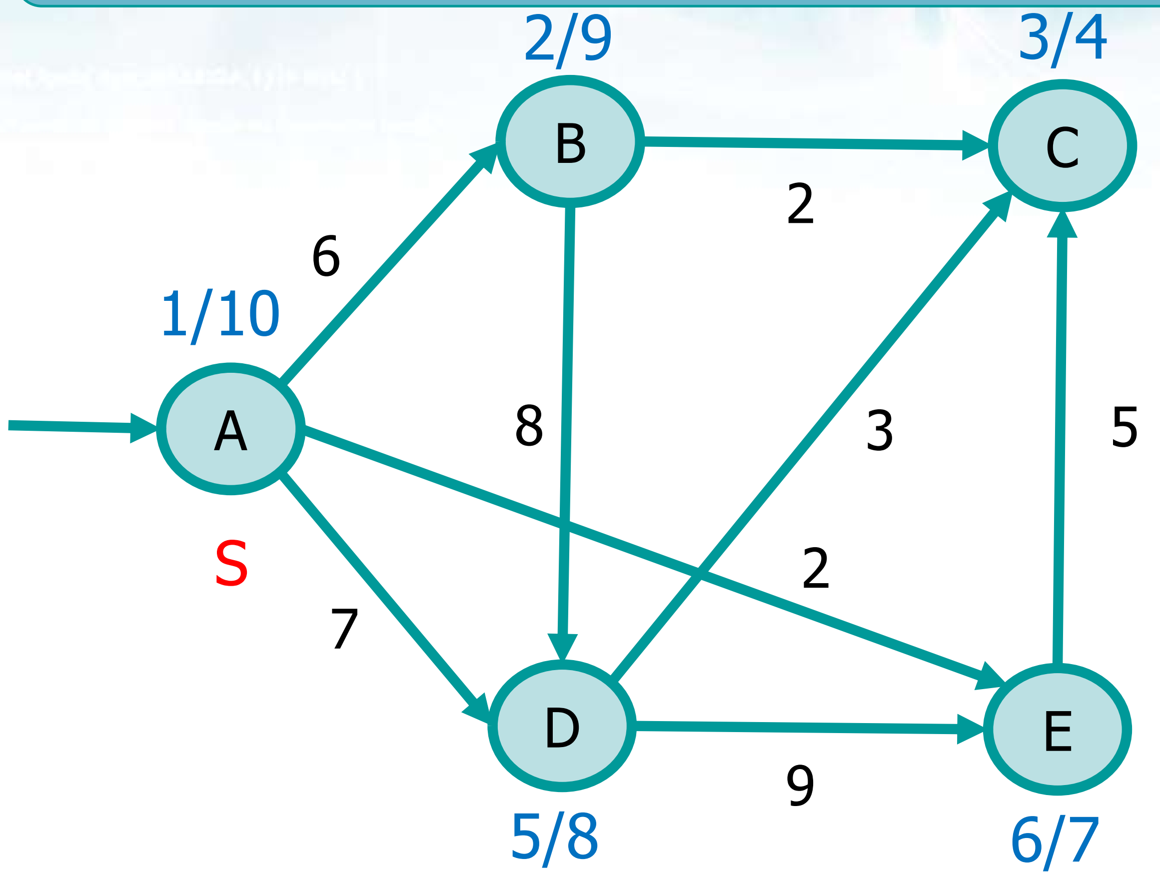
```
v.dist = unchanged
       = 6
v.pred = unchanged
```

Example



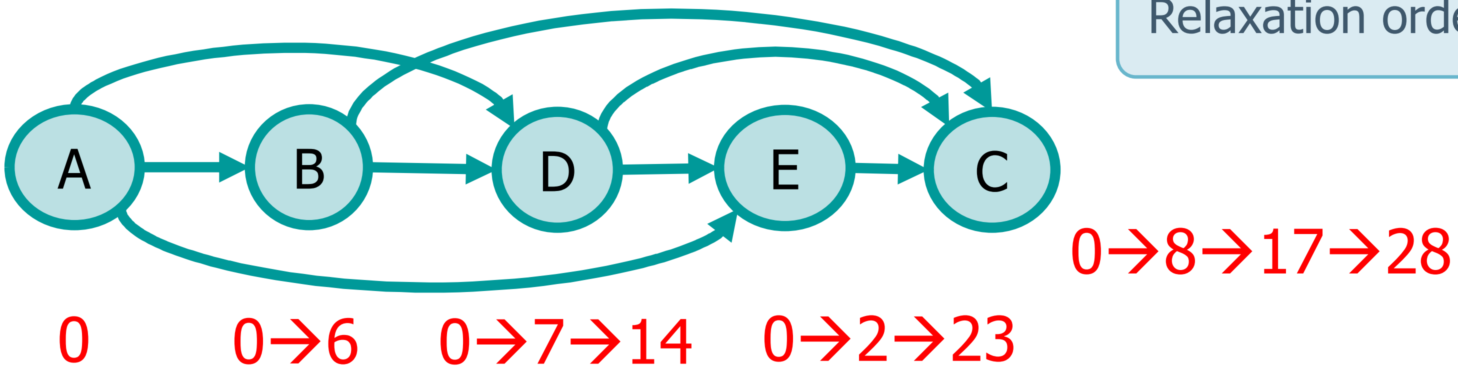
The initial estimate is equal to zero for all vertices

Solution



Relaxation order
(A, B)
(A, D)
(A, E)
(B, D)
(B, C)
(D, E)
(D, C)
(E, C)

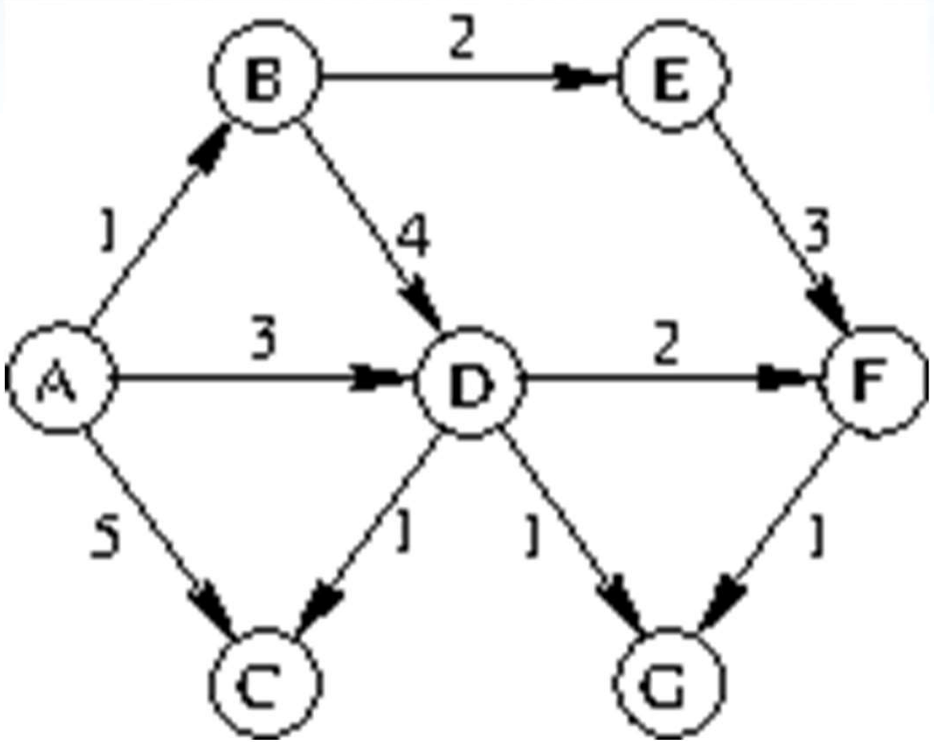
Relaxation order



Complexity

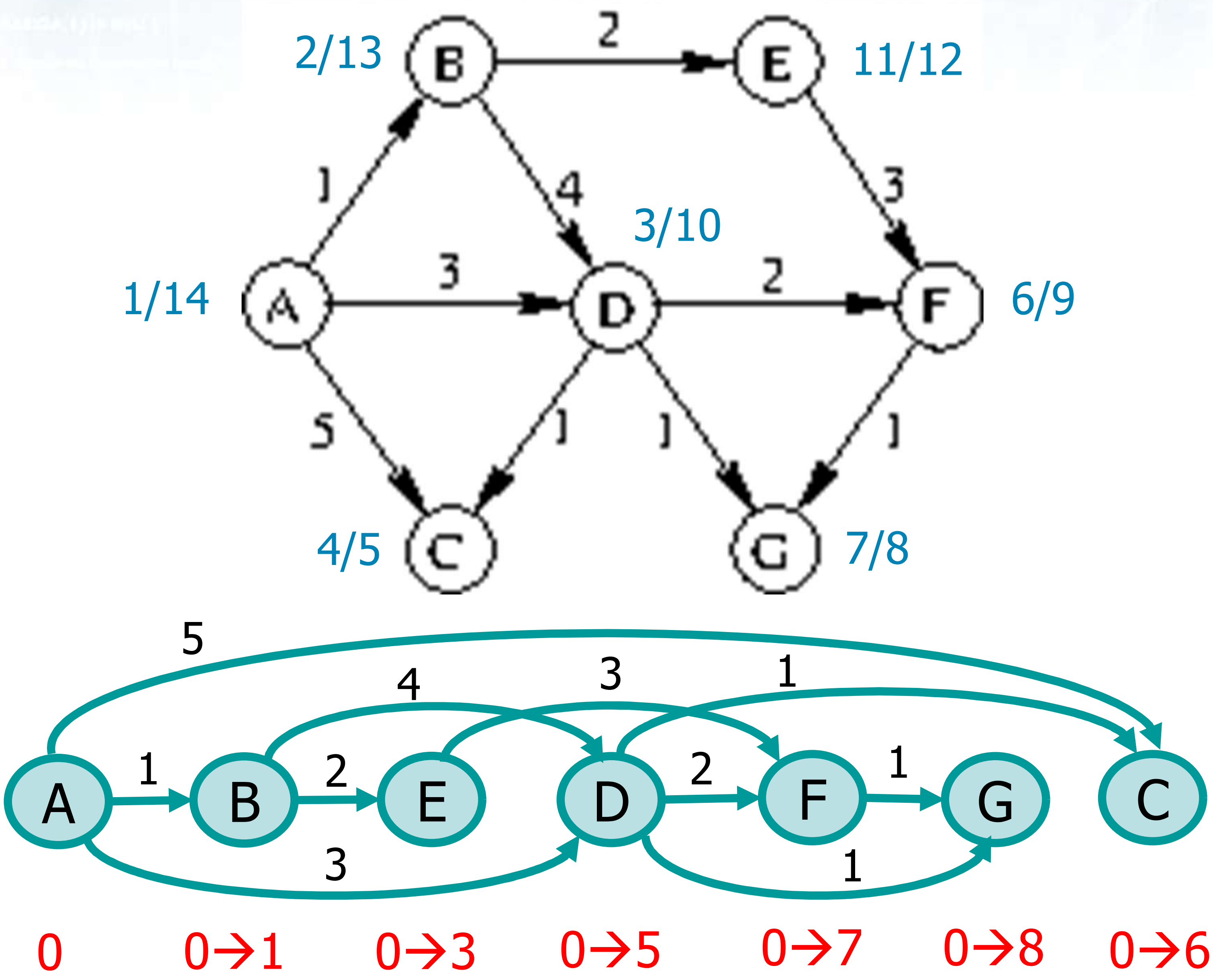
- As the algorithm for the shortest paths for DAGs

Exercise

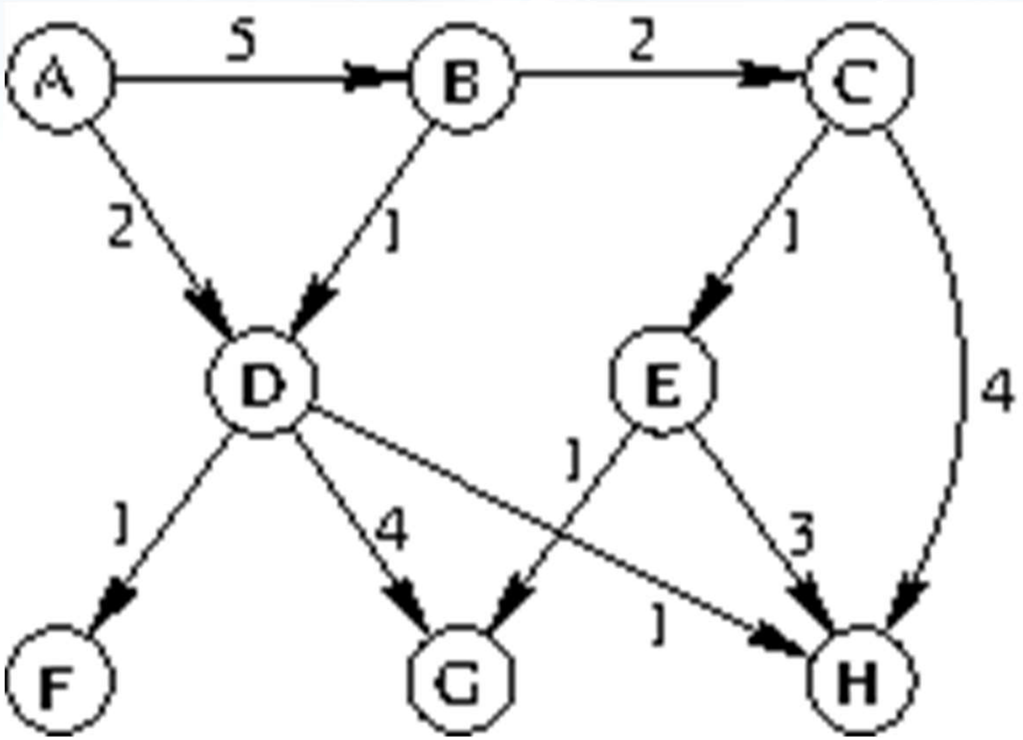


- Given the following graph find all longest paths starting from vertex A

Solution

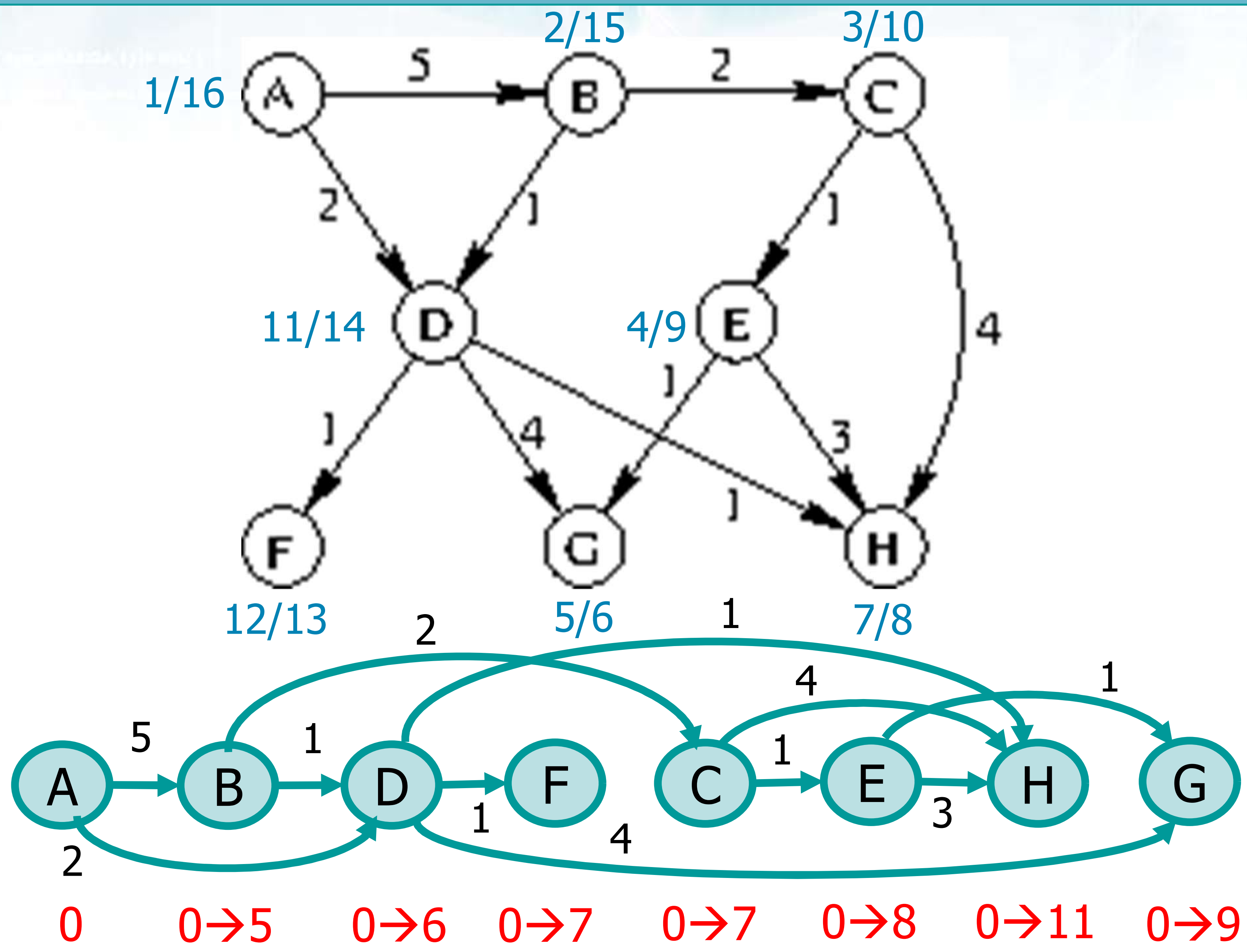

2/13
B
2
E
11/12
1
4
3
3/10
1/14
A
3
D
2
F
6/9
5
1
1
1
4/5
C
G
7/8
5
4
3
1
A
1
B
2
E
D
2
F
1
G
C
3
1
0
0→1
0→3
0→5
0→7
0→8
0→6

Exercise



- Given the following graph find all longest paths starting from vertex A

Solution



2/15
3/10
1/16
A
5
B
2
C
2
1
1
11/14
D
4/9
E
4
1
1
4
3
1
F
G
H
12/13
5/6
7/8
2
1
4
1
5
1
1
A
B
D
F
C
E
H
G
1
4
3
2
0
0→5
0→6
0→7
0→7
0→8
0→11
0→9