

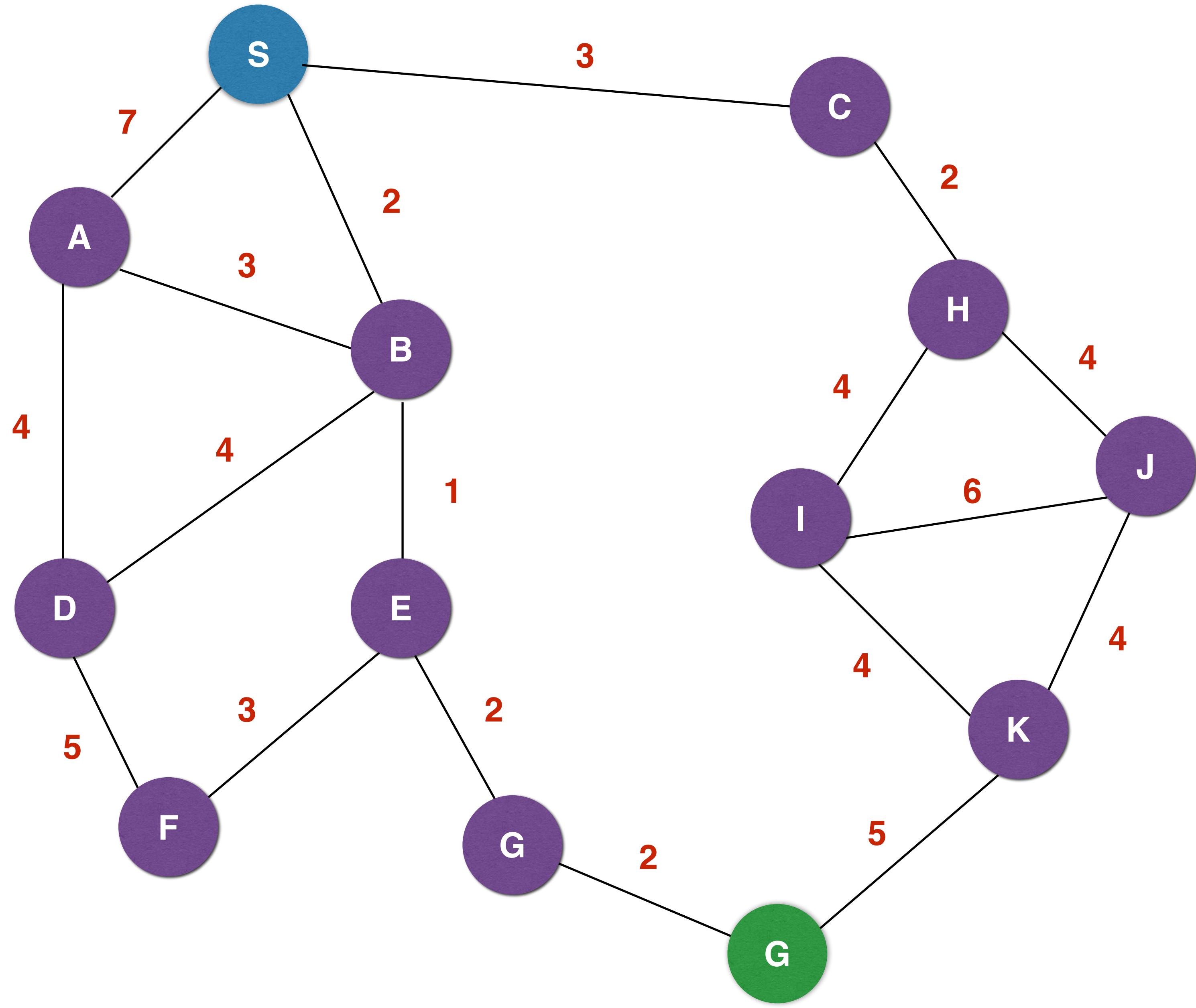
# Path Planning

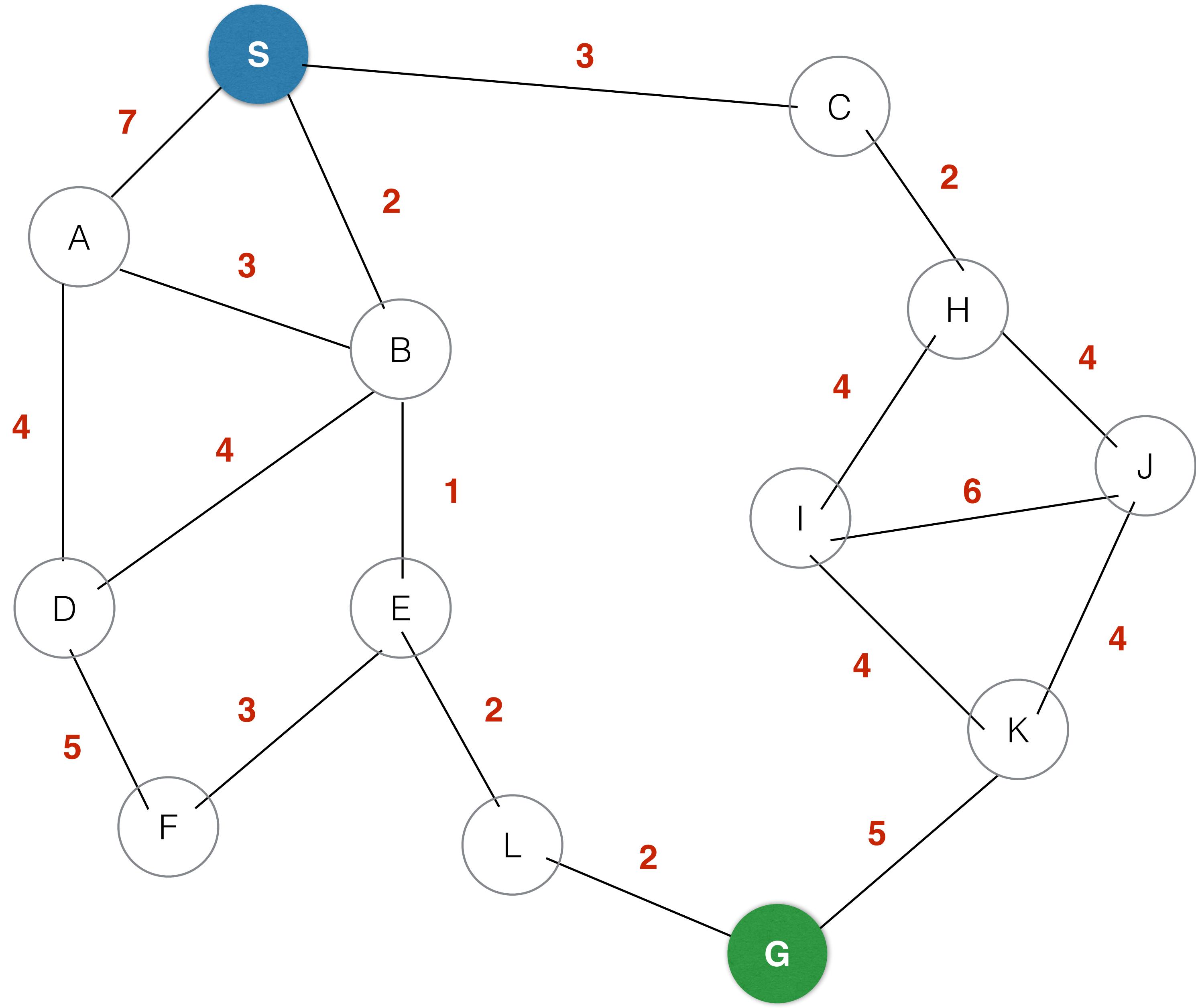
ENAE 380 Flight Software Systems

Lecture 14

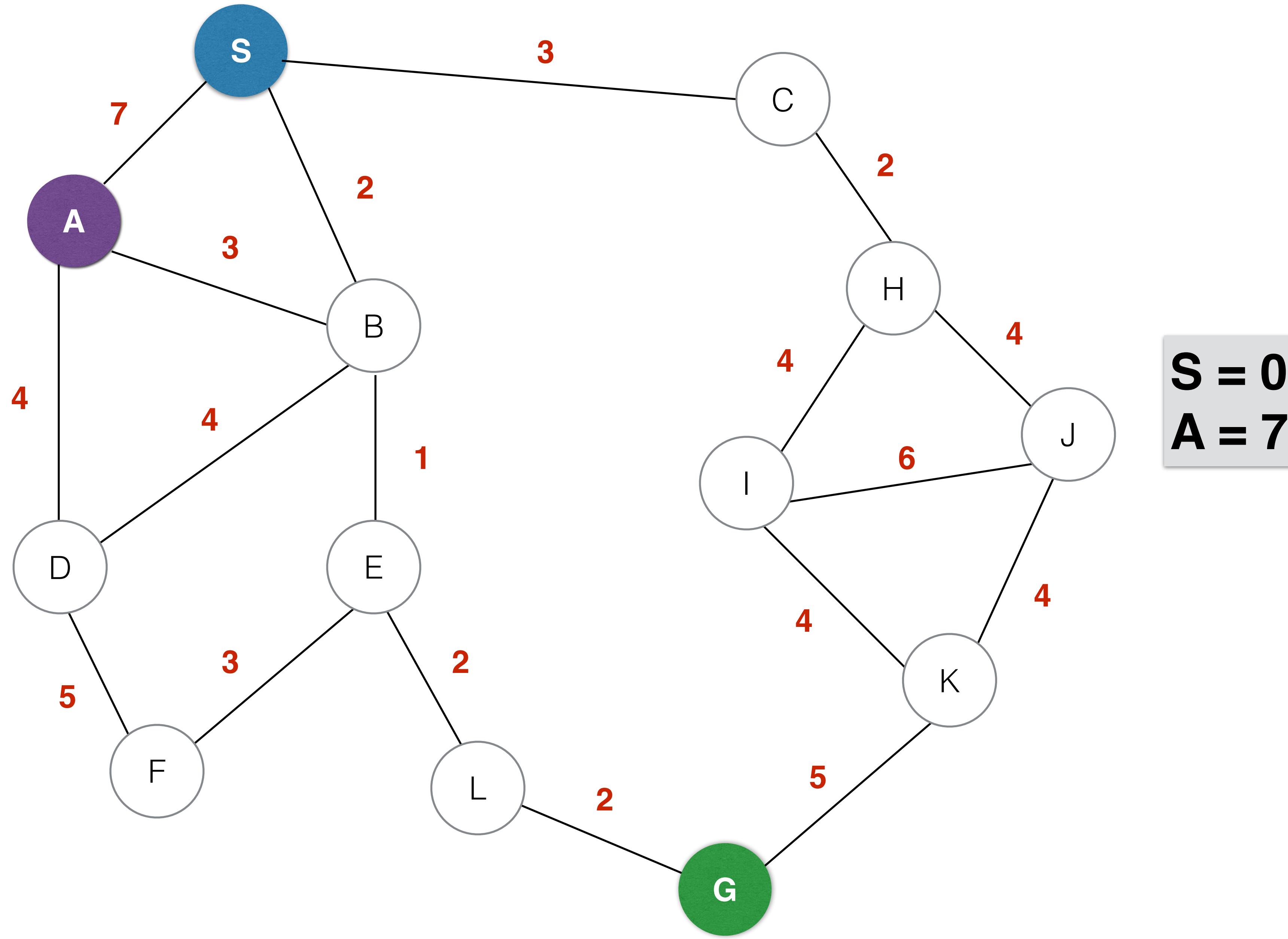
October 28, 2024

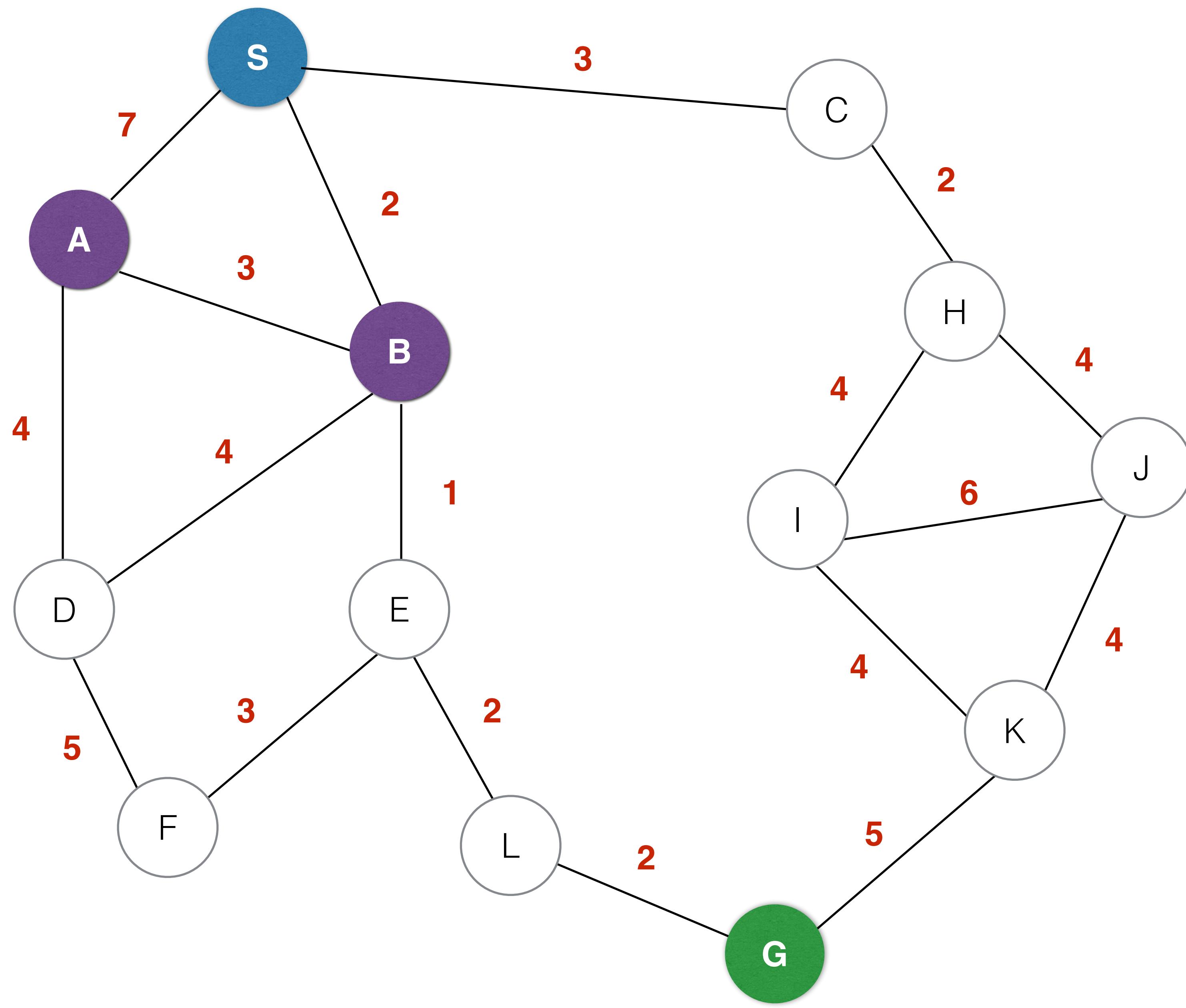
# Dijkstra's Algorithm



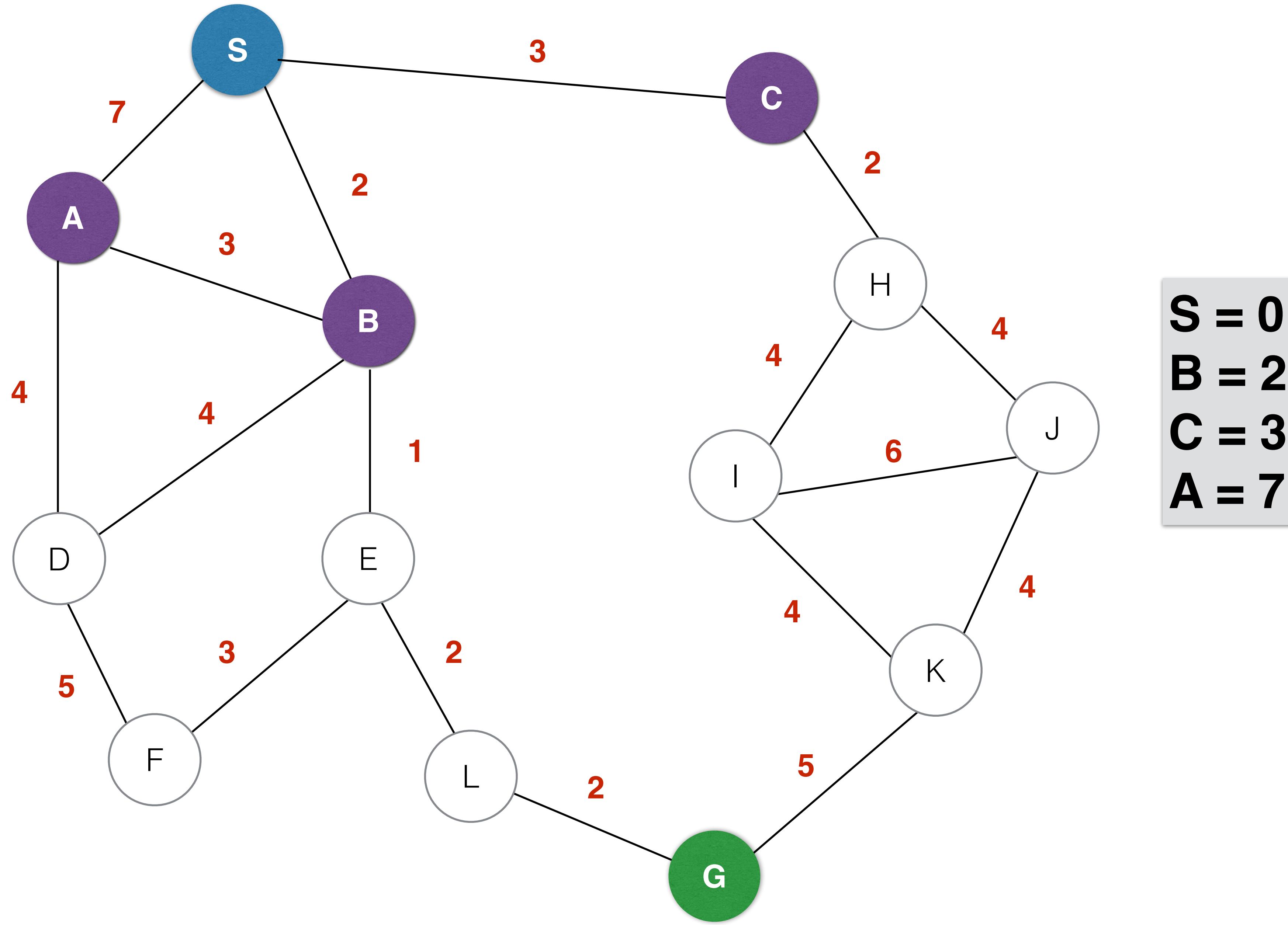


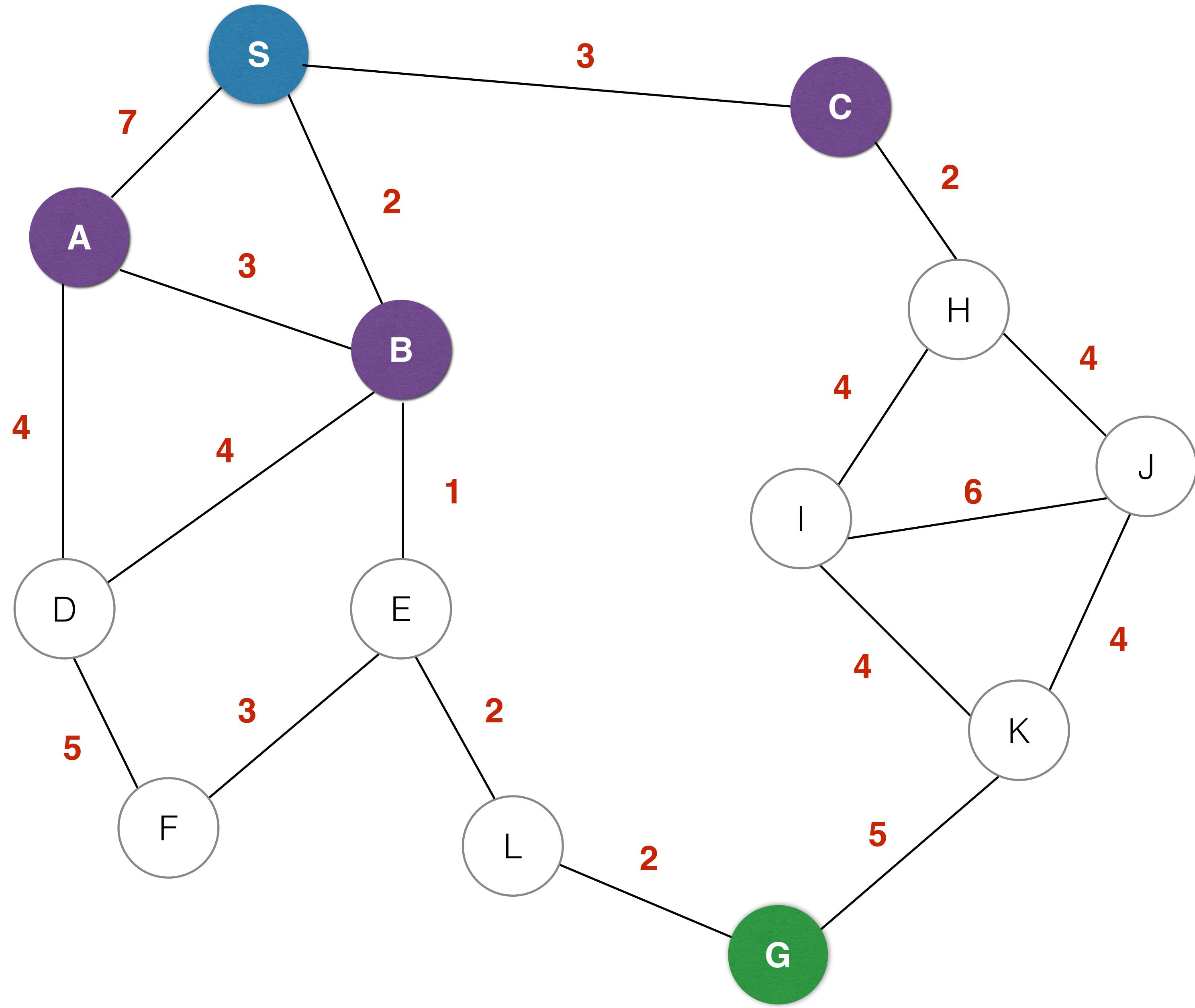
**S = 0**



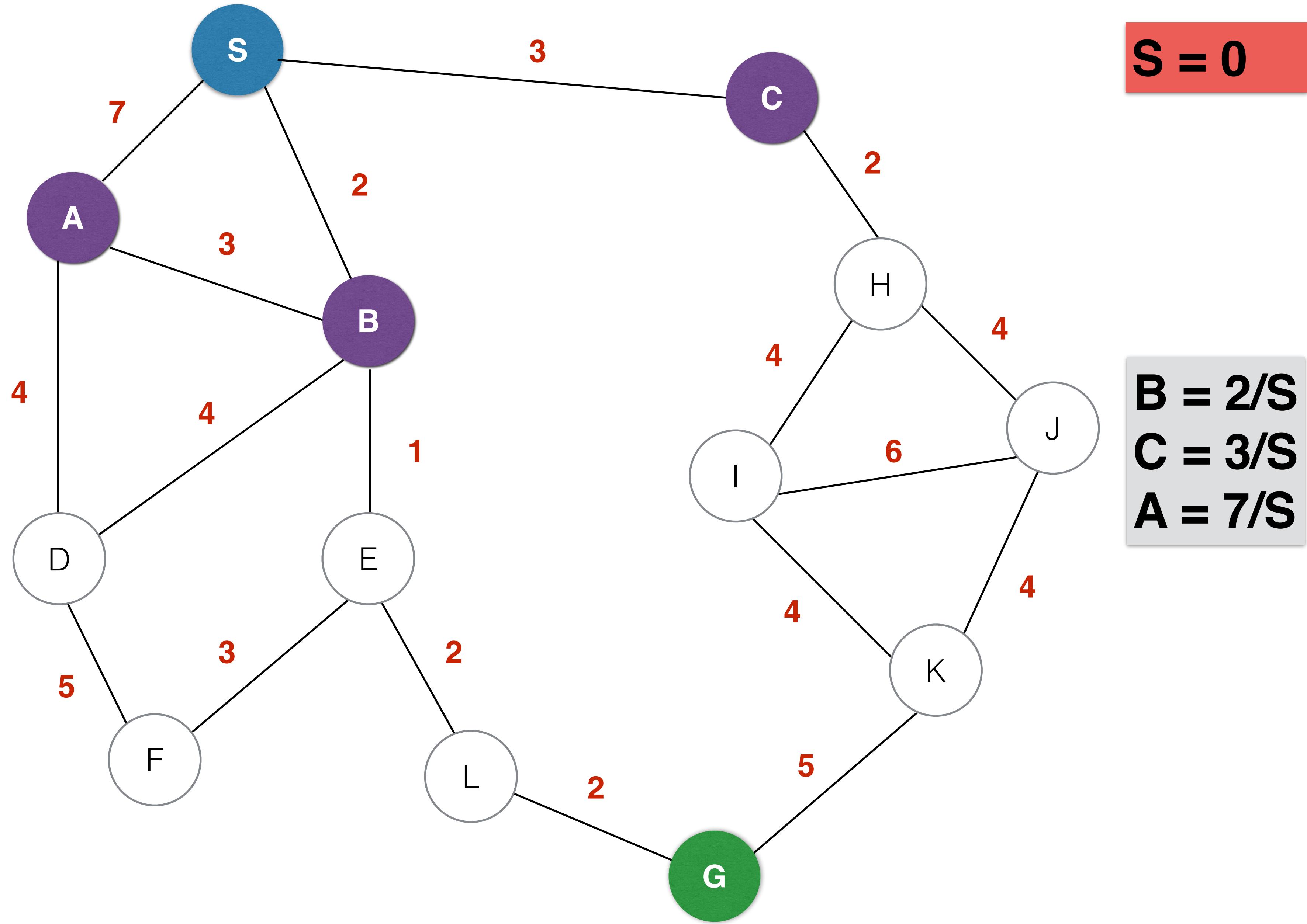


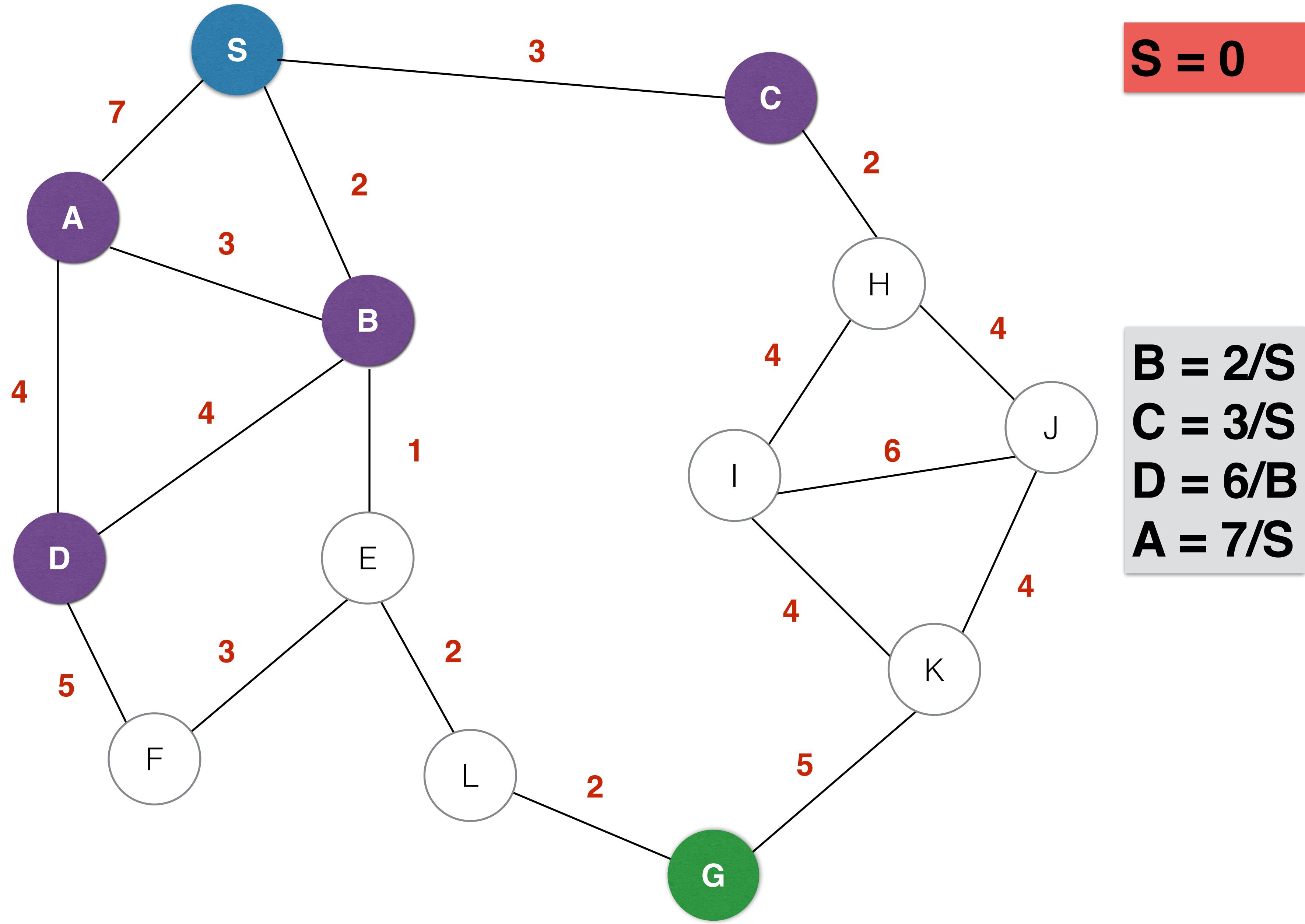
**S = 0  
B = 2  
A = 7**

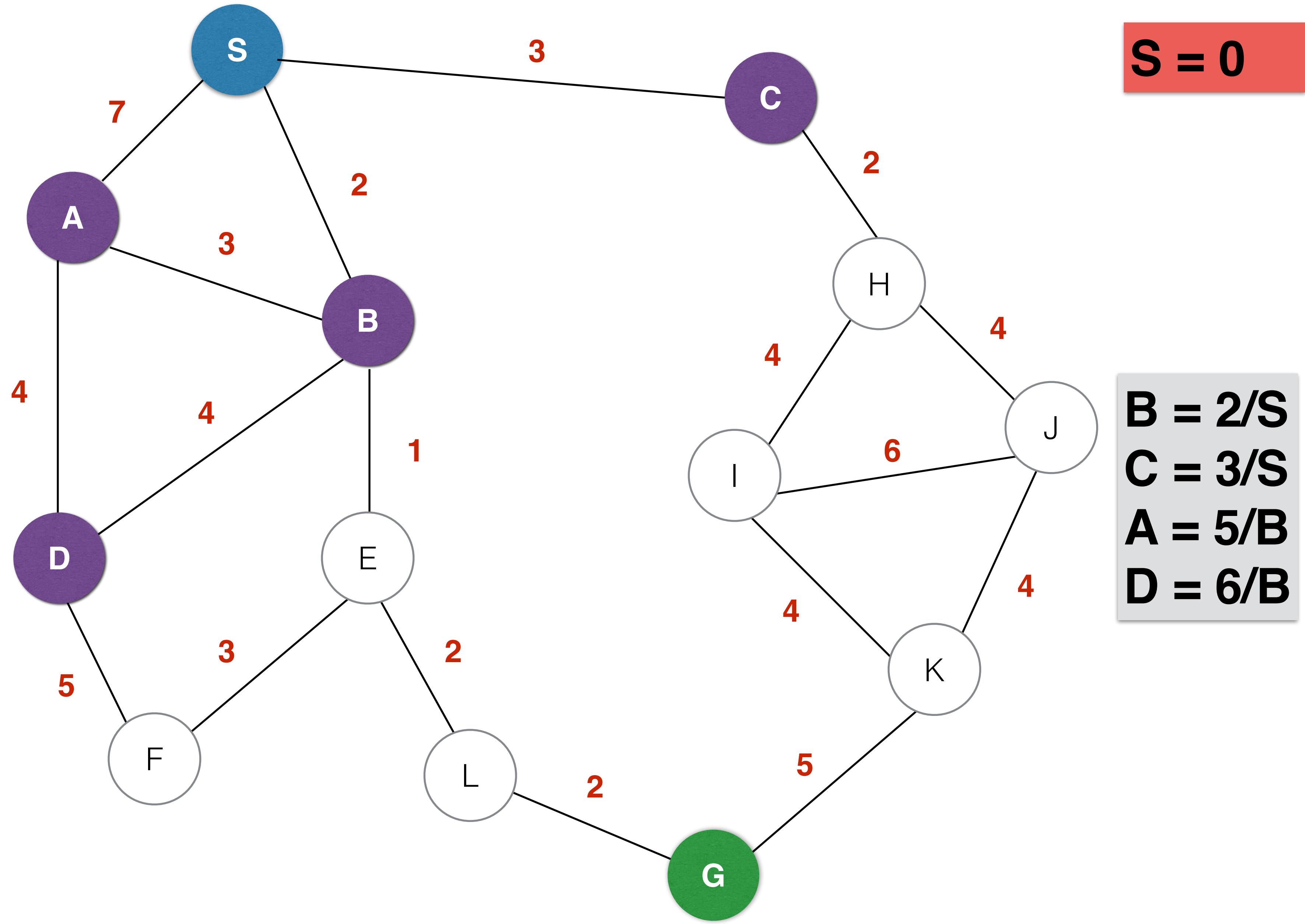


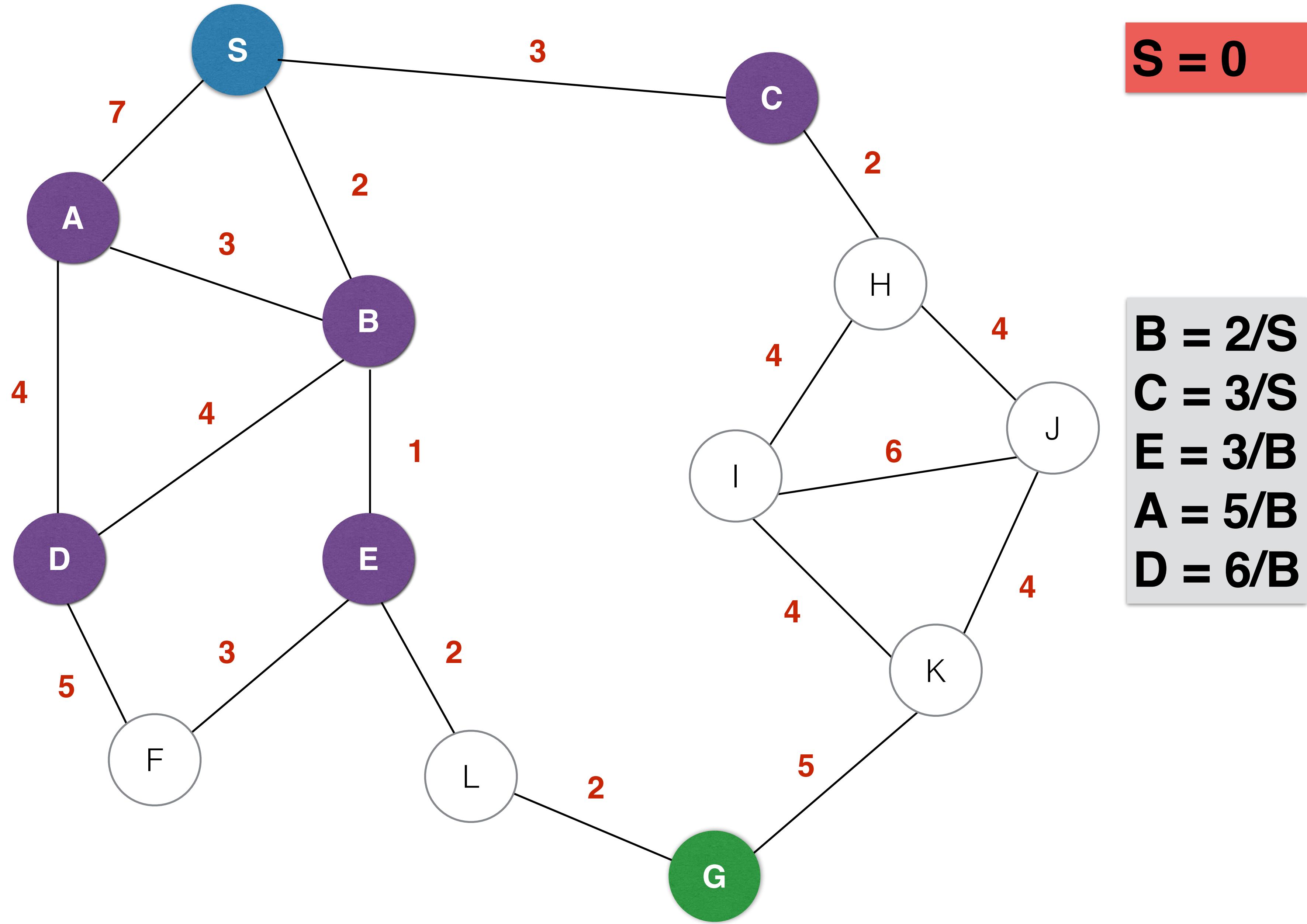


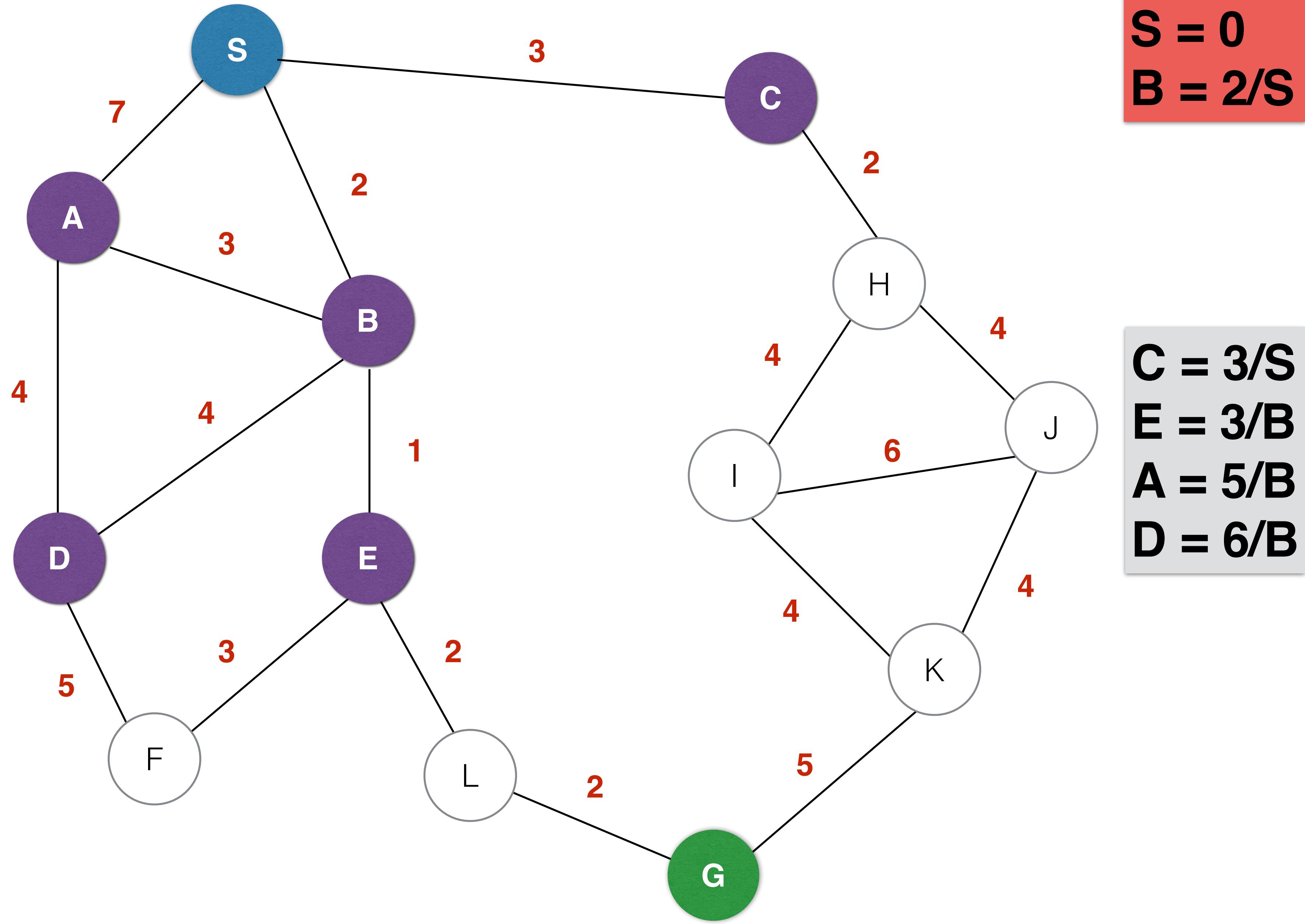
**S = 0**  
**B = 2/S**  
**C = 3/S**  
**A = 7/S**

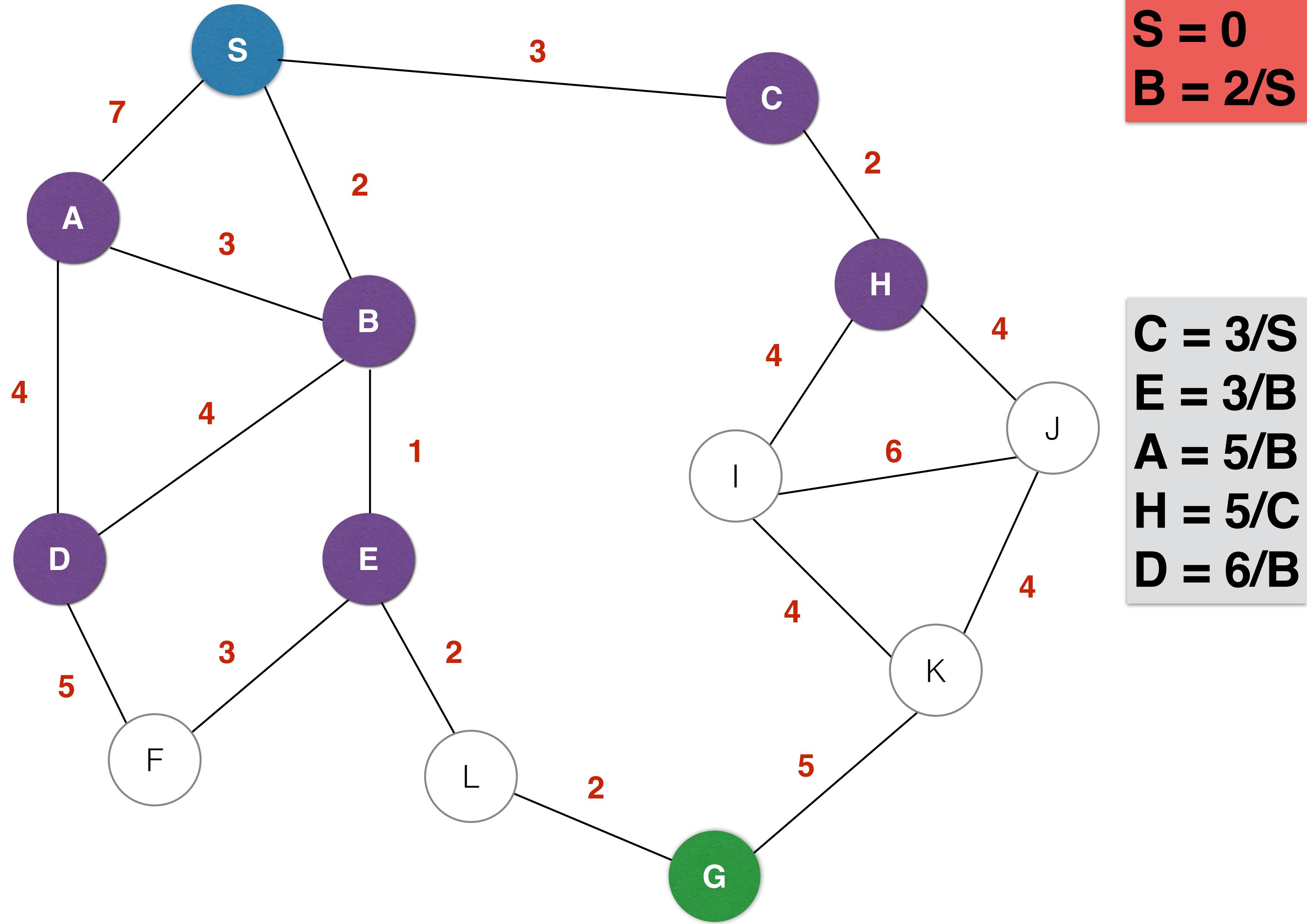


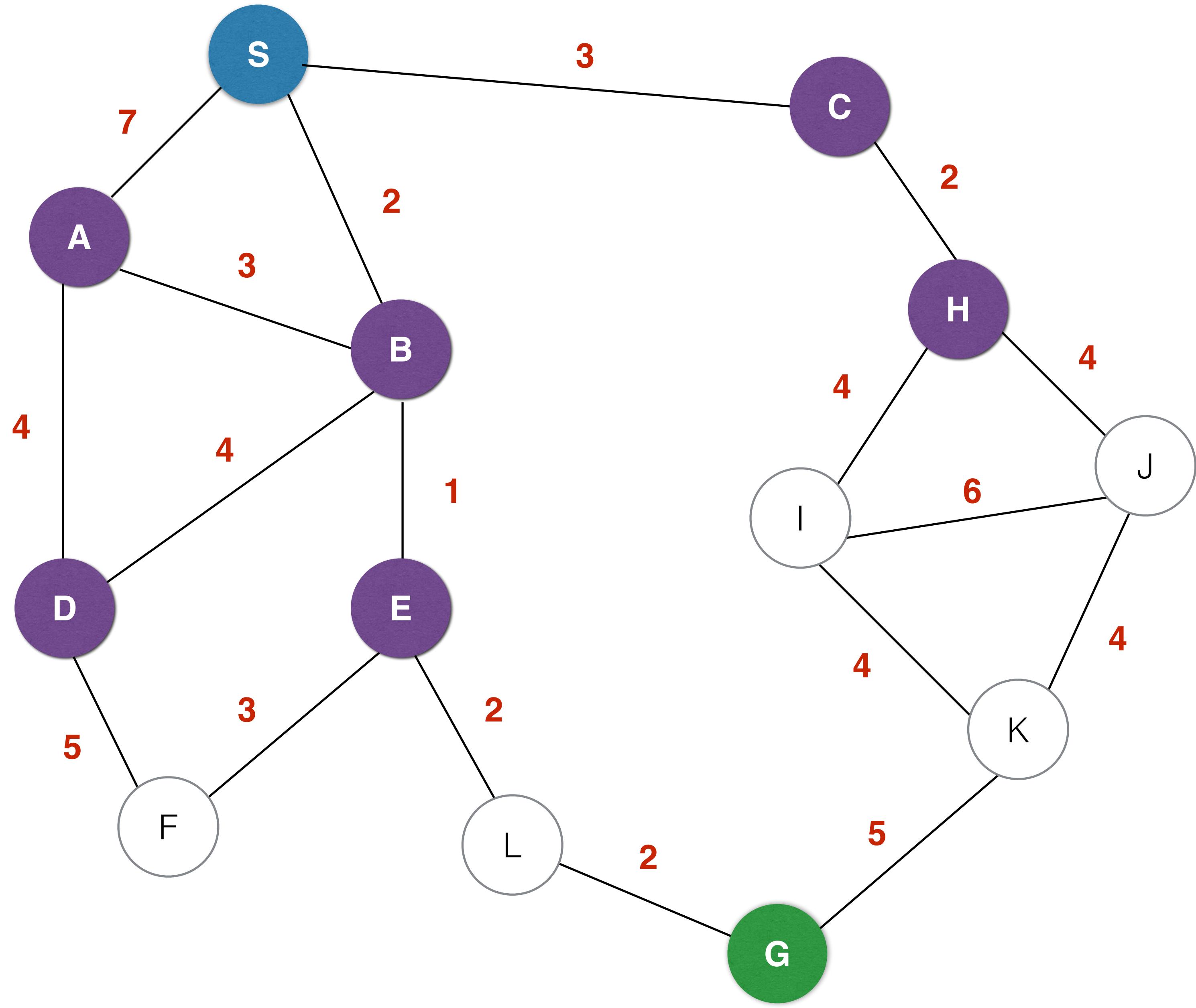






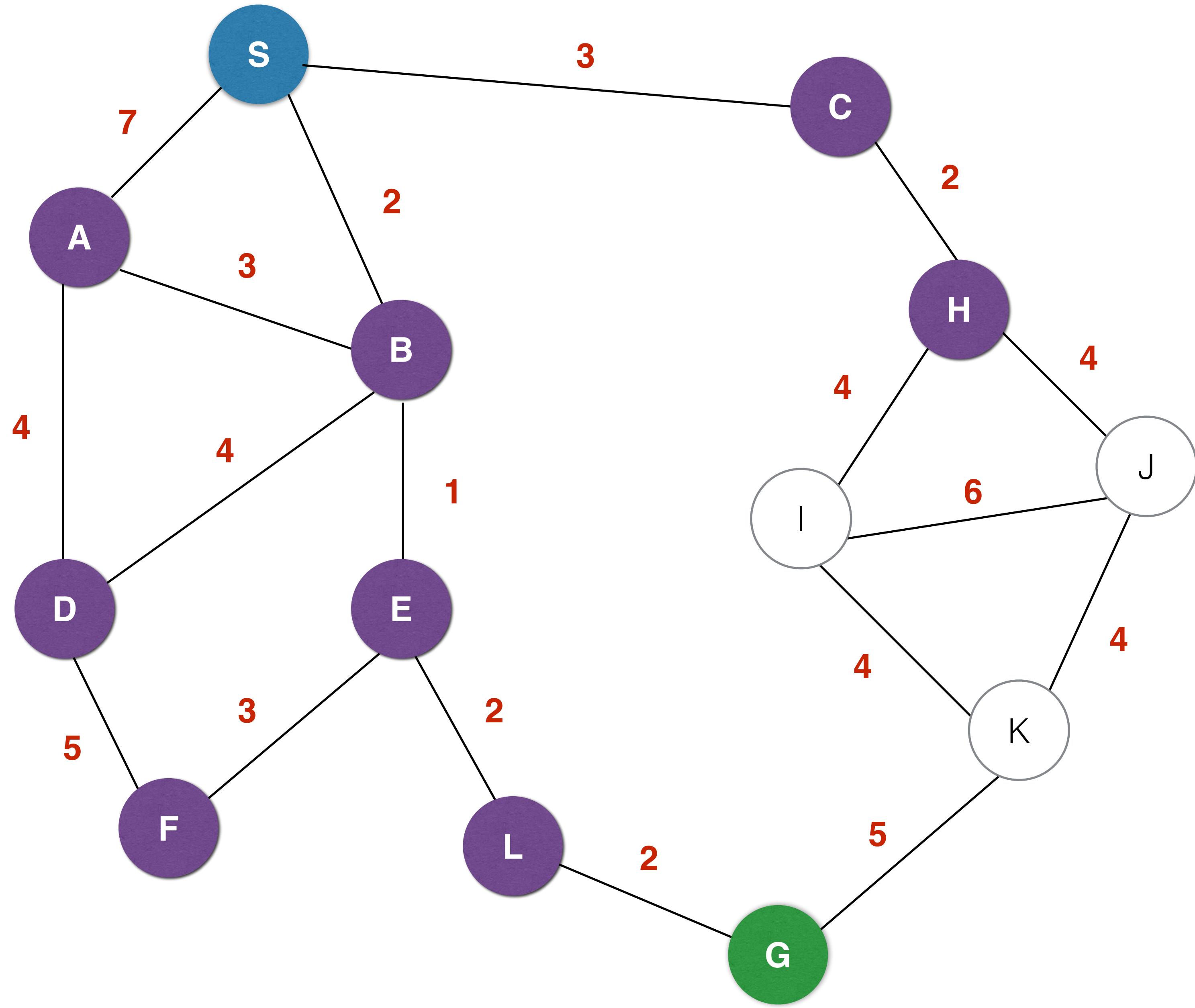






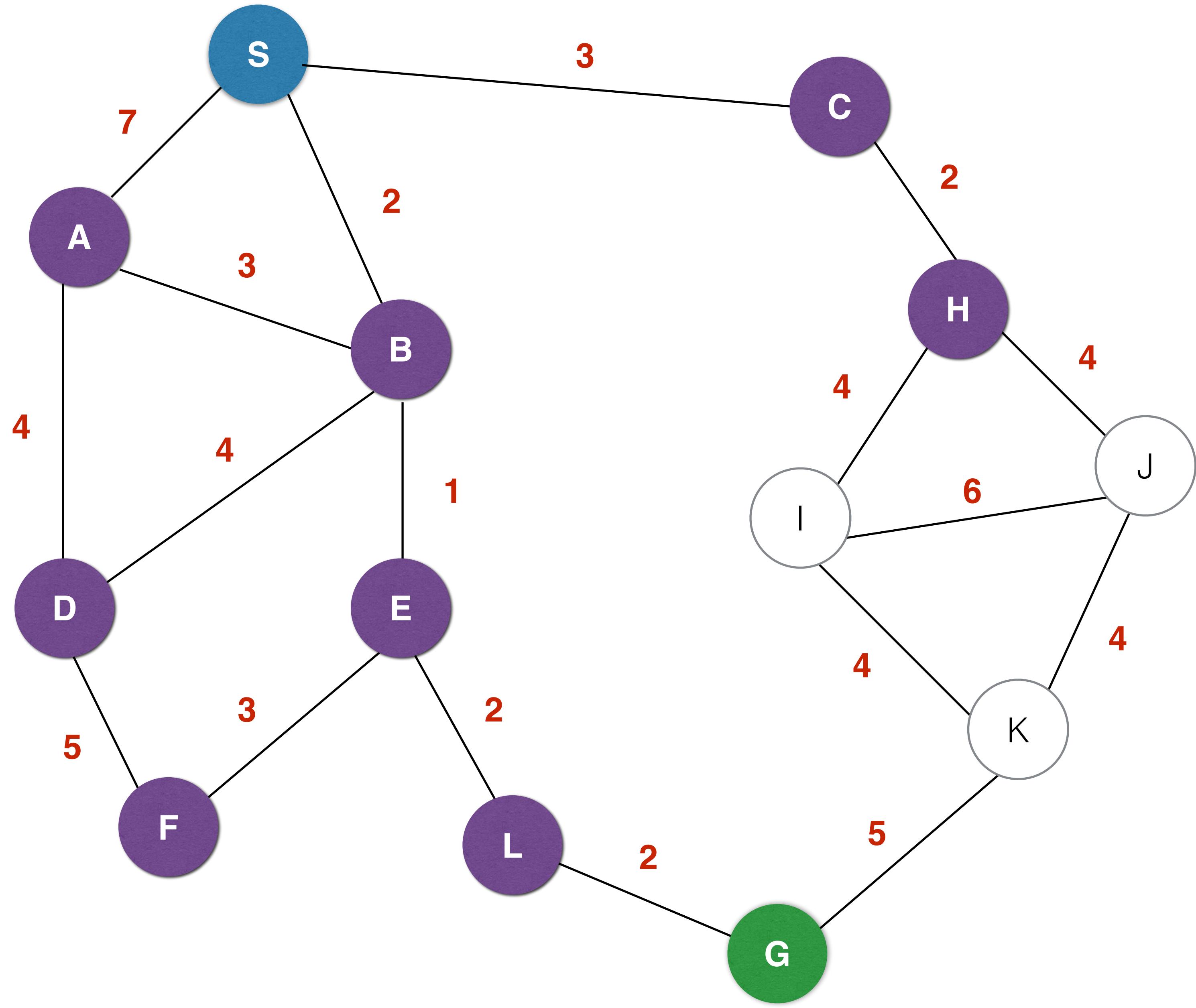
$$\begin{aligned} S &= 0 \\ B &= 2/S \\ C &= 3/S \end{aligned}$$

$$\begin{aligned} E &= 3/B \\ A &= 5/B \\ H &= 5/C \\ D &= 6/B \end{aligned}$$



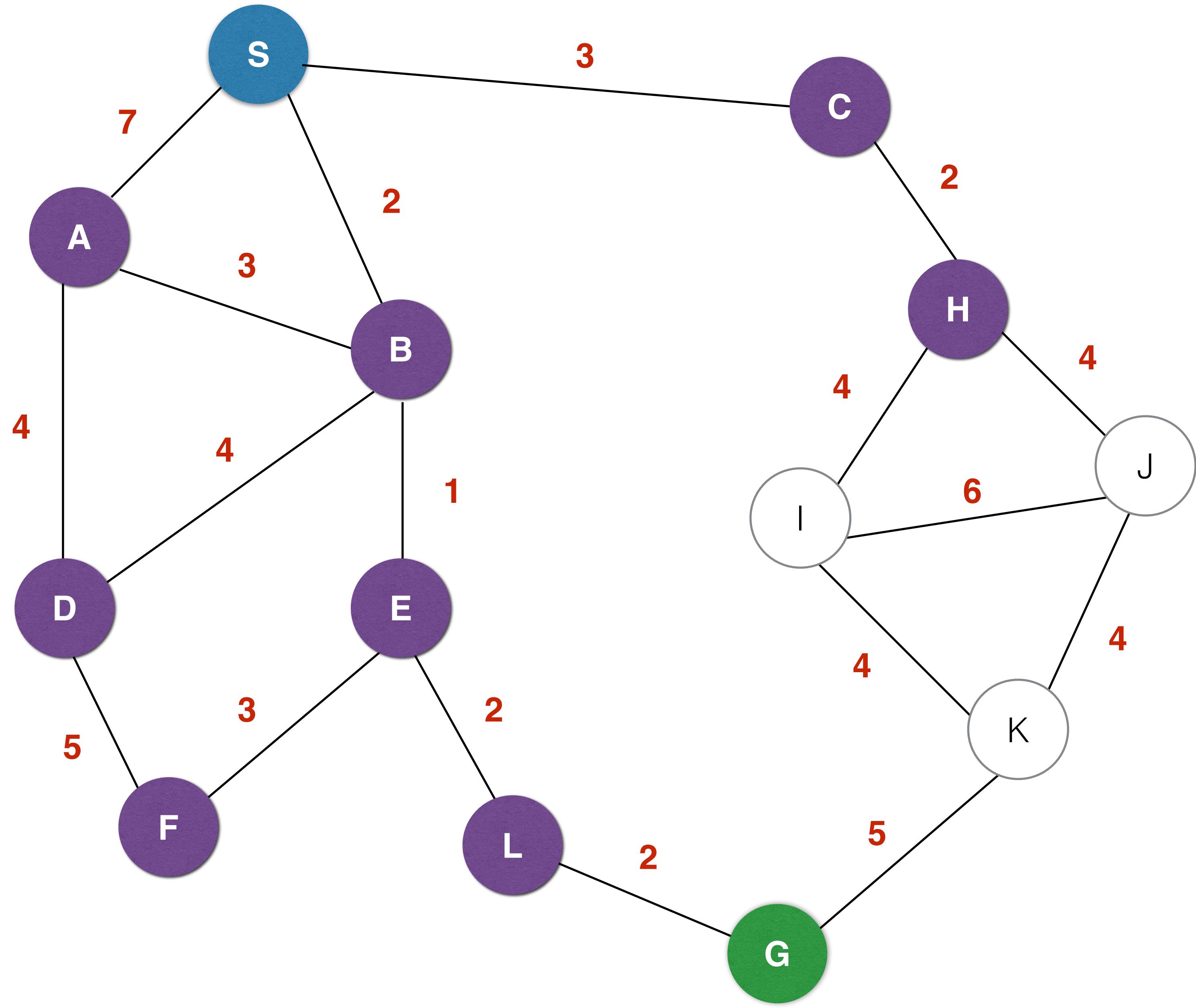
$$\begin{aligned}S &= 0 \\B &= 2/S \\C &= 3/S\end{aligned}$$

$$\begin{aligned}E &= 3/B \\A &= 5/B \\H &= 5/C \\L &= 5/E \\D &= 6/B \\F &= 6/E\end{aligned}$$



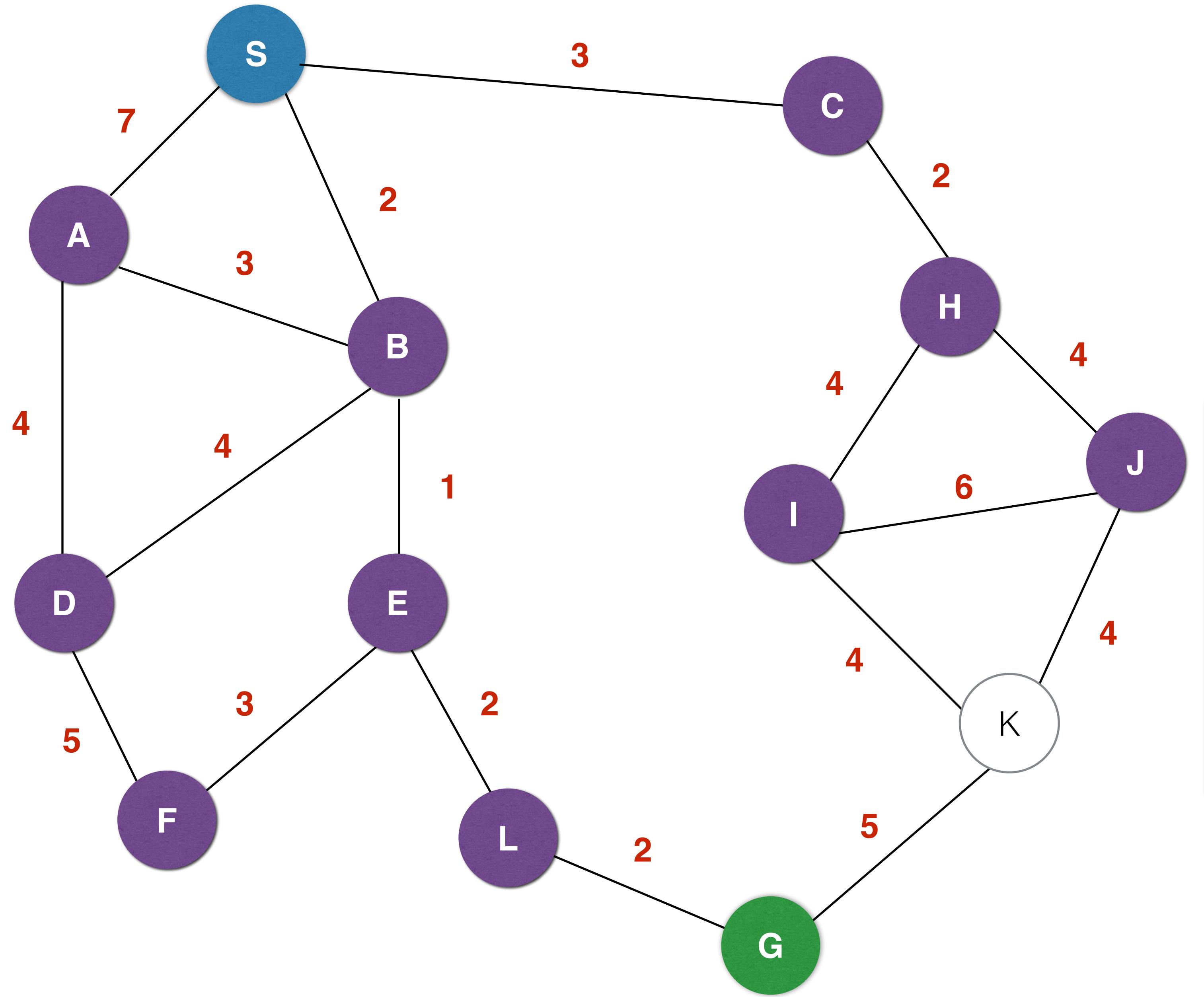
$$\begin{aligned} S &= 0 \\ B &= 2/S \\ C &= 3/S \\ E &= 3/B \end{aligned}$$

$$\begin{aligned} A &= 5/B \\ H &= 5/C \\ L &= 5/E \\ D &= 6/B \\ F &= 6/E \end{aligned}$$



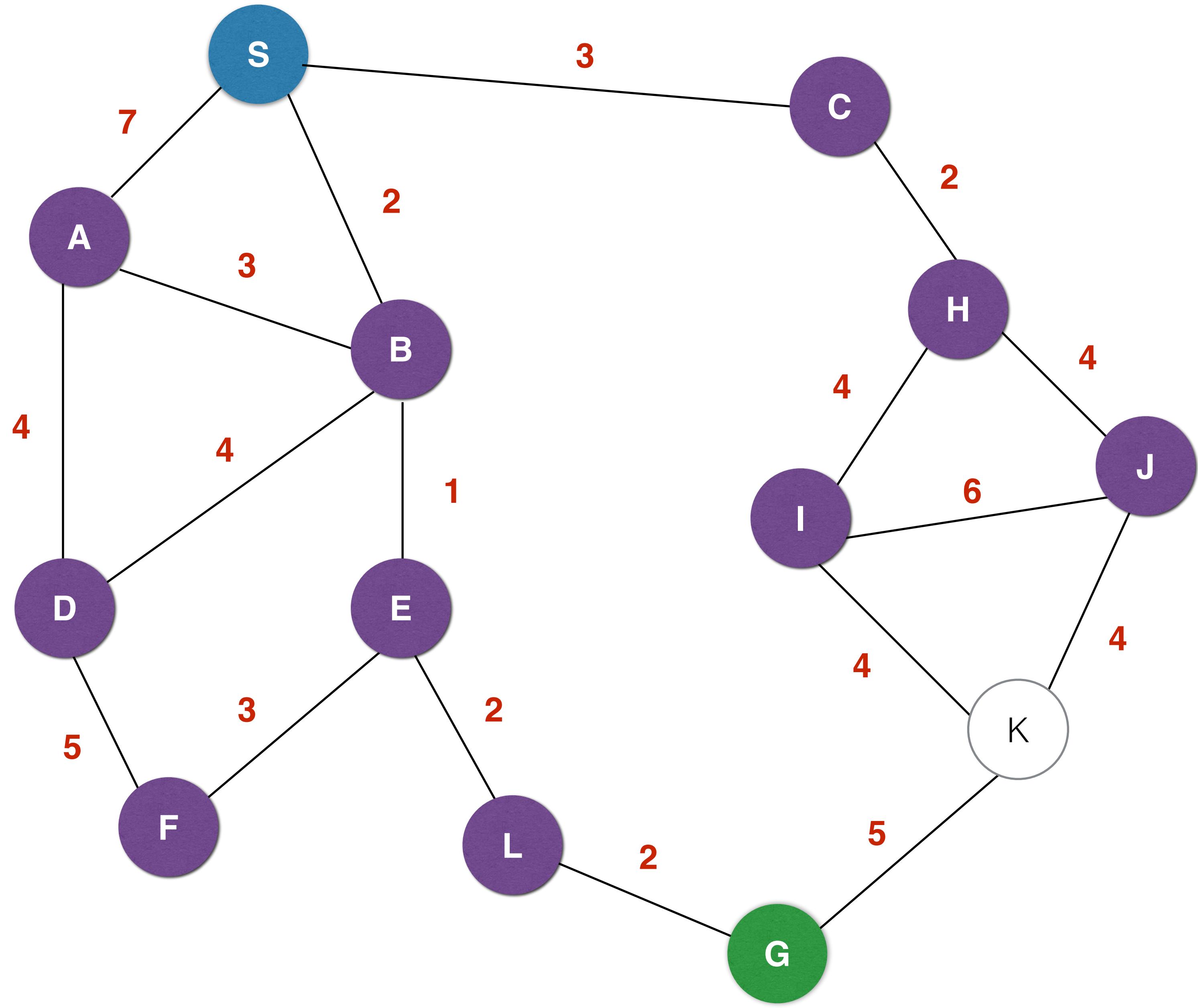
$$\begin{aligned}
 S &= 0 \\
 B &= 2/S \\
 C &= 3/S \\
 E &= 3/B \\
 A &= 5/B
 \end{aligned}$$

$$\begin{aligned}
 H &= 5/C \\
 L &= 5/E \\
 D &= 6/B \\
 F &= 6/E
 \end{aligned}$$



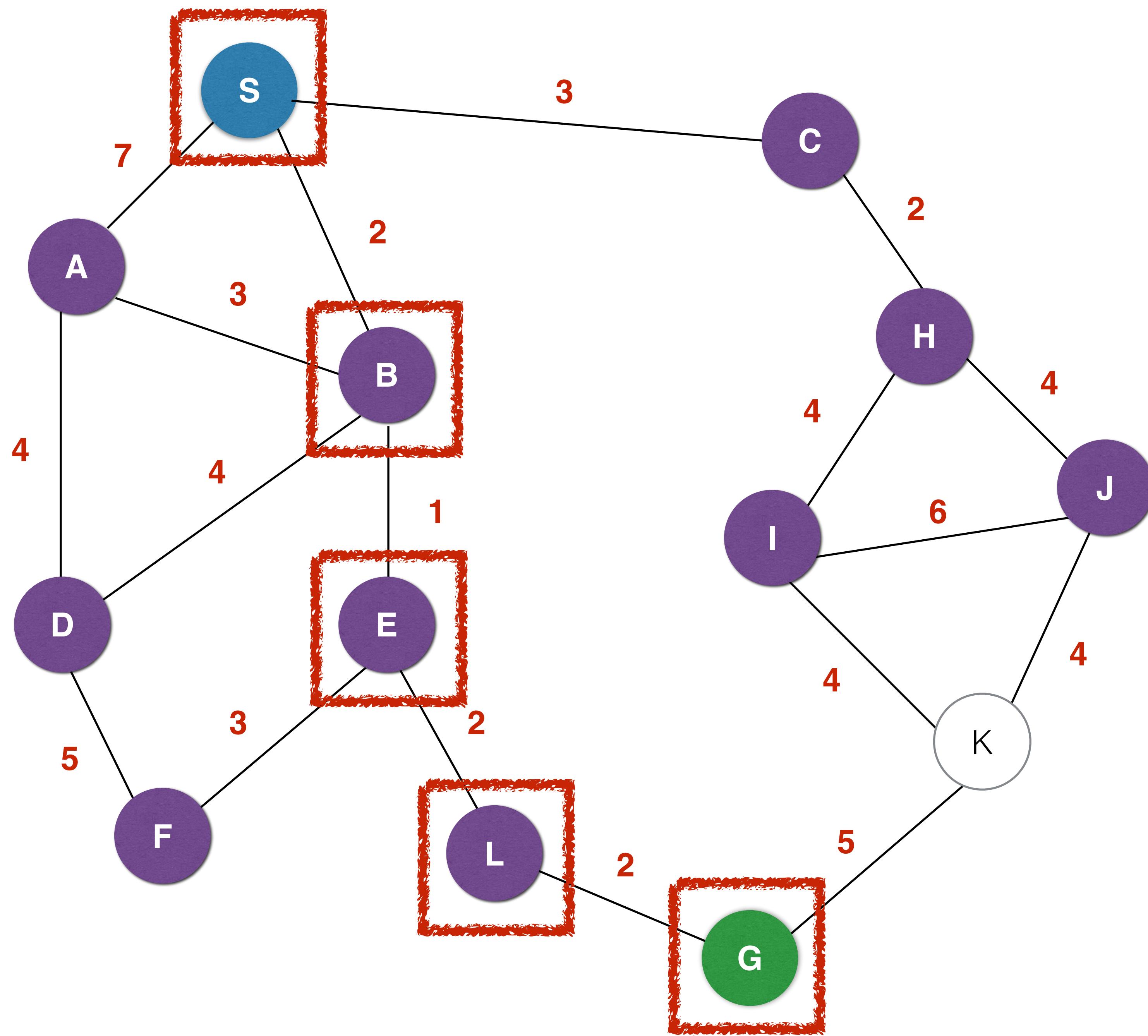
$$\begin{aligned}S &= 0 \\B &= 2/S \\C &= 3/S \\E &= 3/B \\A &= 5/B\end{aligned}$$

$$\begin{aligned}H &= 5/C \\L &= 5/E \\D &= 6/B \\F &= 6/E \\I &= 9/H \\J &= 9/H\end{aligned}$$



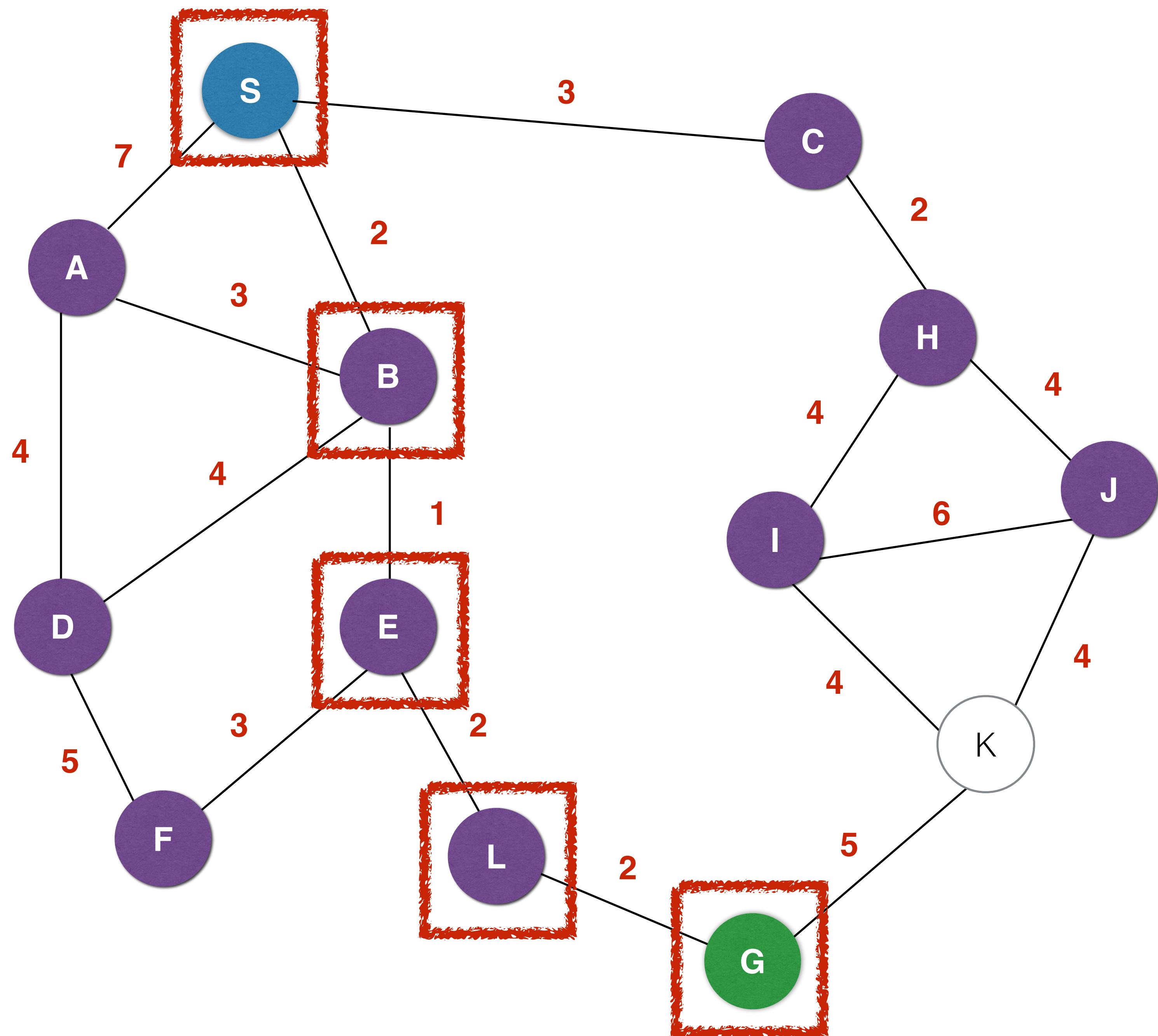
$S = 0$
$B = 2/S$
$C = 3/S$
$E = 3/B$
$A = 5/B$
$H = 5/C$

$L = 5/E$
$D = 6/B$
$F = 6/E$
$G = 7/L$
$I = 9/H$
$J = 9/H$
$K = \inf$



$S = 0$
$B = 2/S$
$C = 3/S$
$E = 3/B$
$A = 5/B$
$H = 5/C$

$L = 5/E$
$D = 6/B$
$F = 6/E$
$G = 7/L$
$I = 9/H$
$J = 9/H$
$K = \inf$



$$\begin{aligned}
 S &= 0 \\
 B &= 2/S \\
 C &= 3/S \\
 E &= 3/B \\
 A &= 5/B \\
 H &= 5/C
 \end{aligned}$$

$$G = 7/L$$

# Planning

## **Robotics**

Motion Planning  
Trajectory Planning

## **Control Theory**

## **Artificial Intelligence**

# Planning

## **Robotics**

Motion Planning  
Trajectory Planning

## **Control Theory**

Feedback Policies  
Stability

## **Artificial Intelligence**

# Planning

## **Robotics**

Motion Planning  
Trajectory Planning

## **Control Theory**

Feedback Policies  
Stability

## **Artificial Intelligence**

Planning/AI Planning  
Discrete Tasks

# Planning

## Robotics

Motion Planning  
Trajectory Planning

Robot/Decision Maker

## Control Theory

Feedback Policies  
Stability

Controller

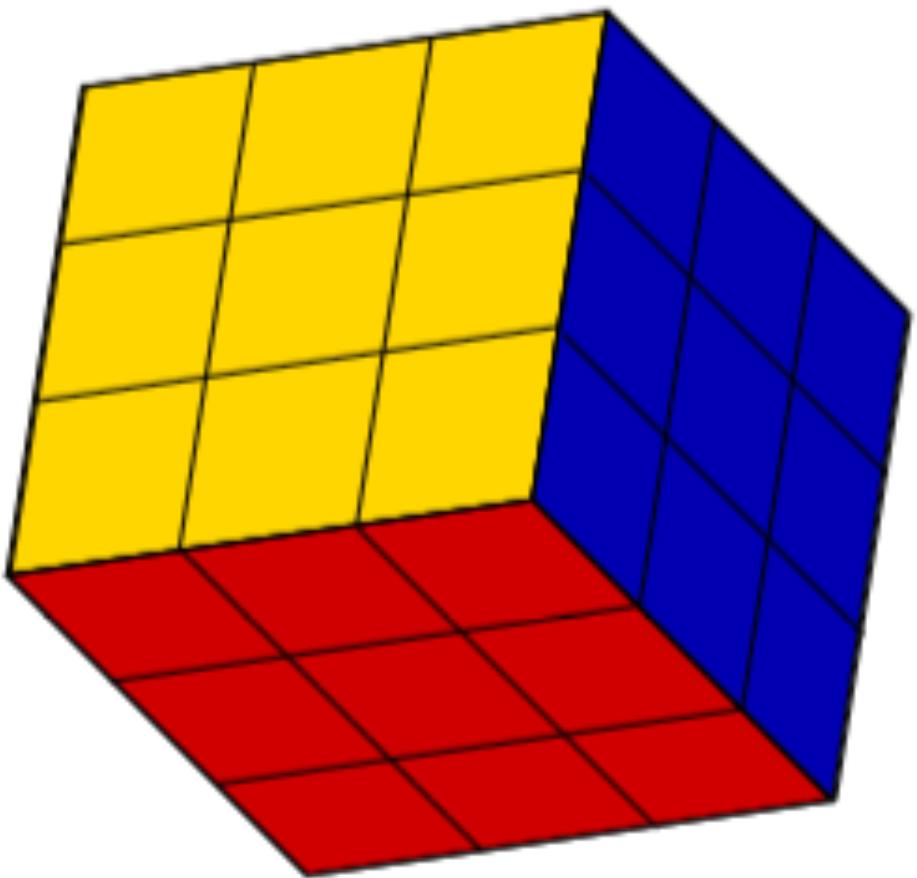
## Artificial Intelligence

Planning/AI Planning  
Discrete Tasks

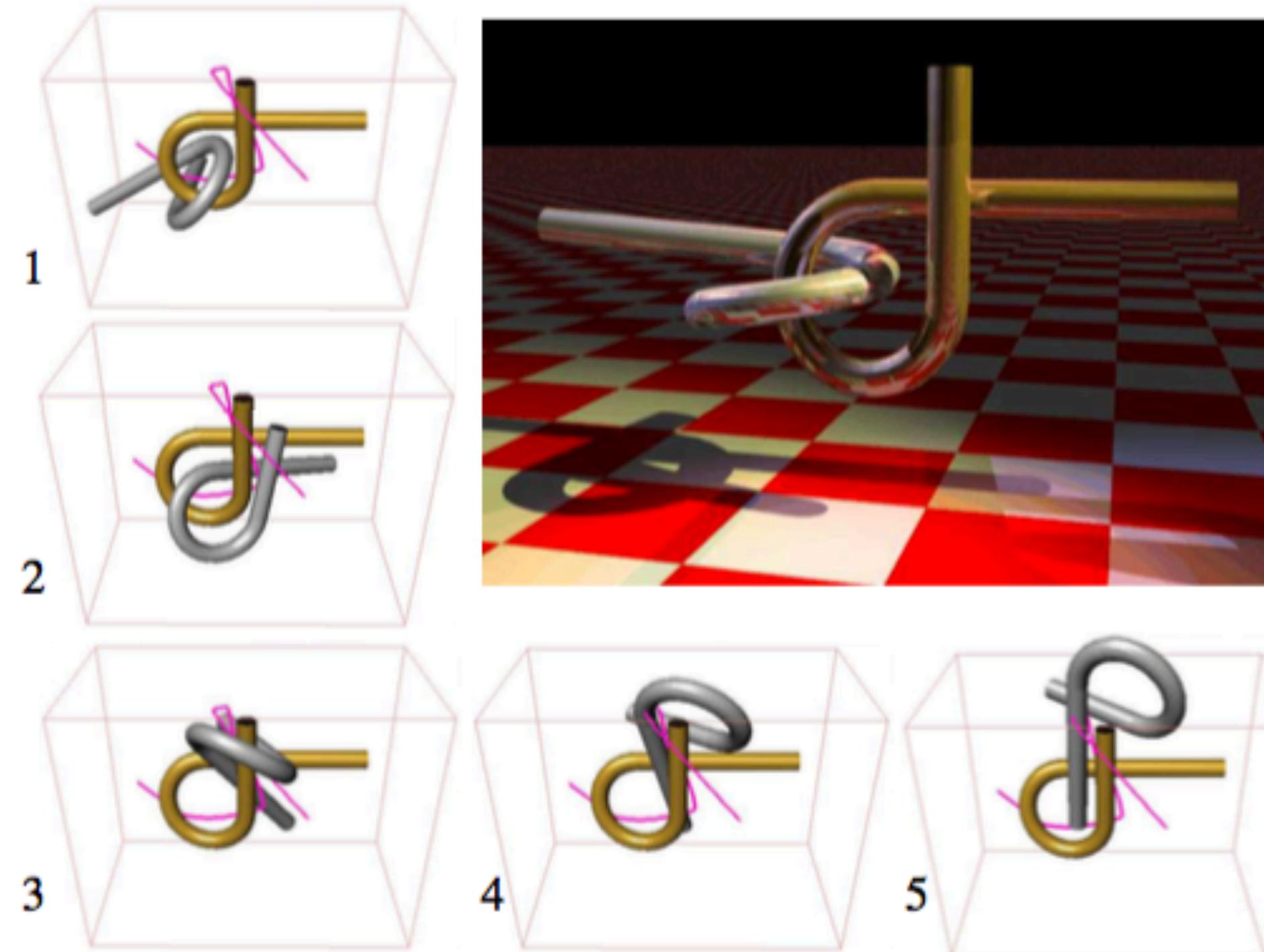
Agent

# Planning Problems

# Motion Planning Puzzle



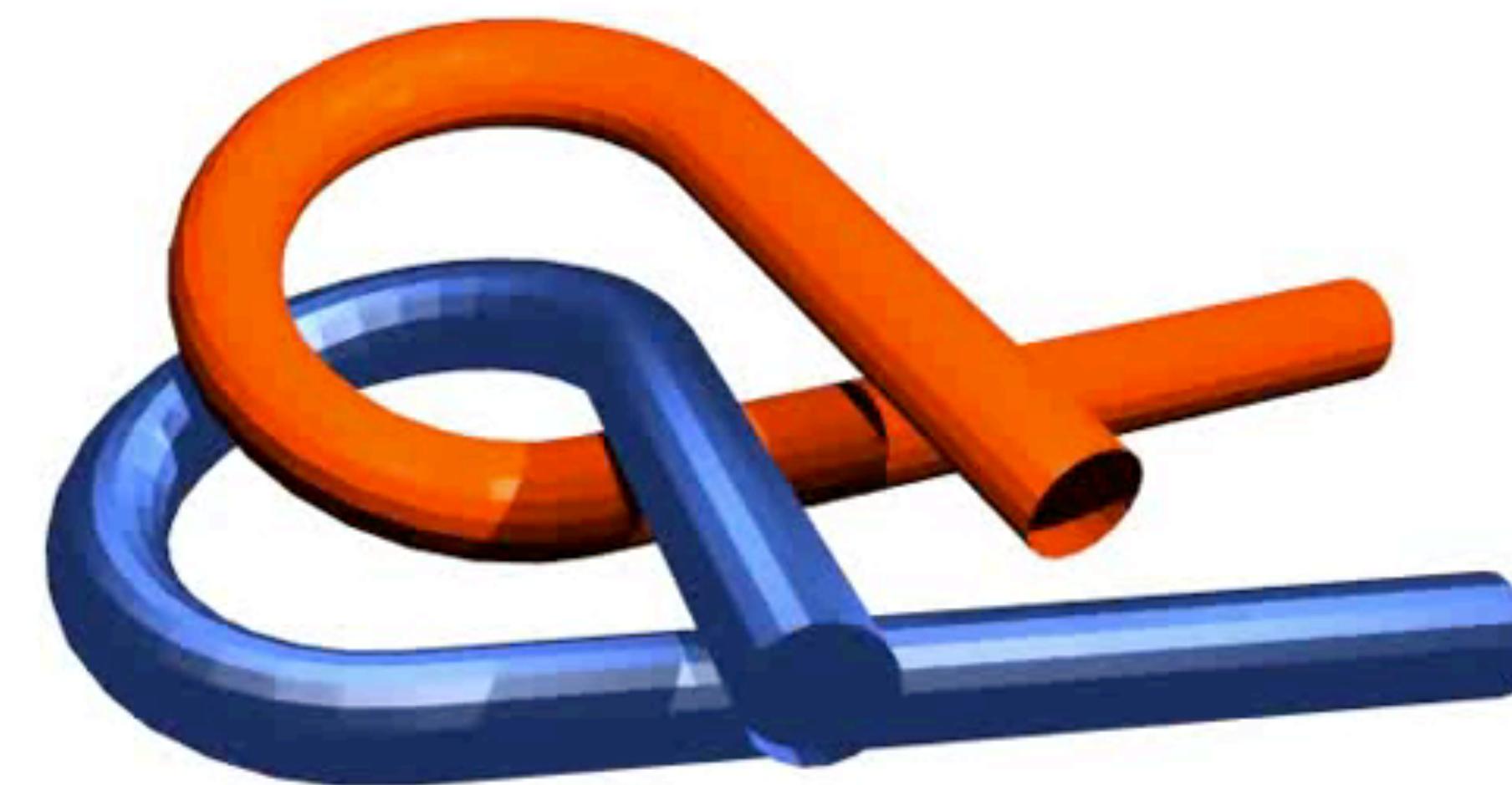
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	



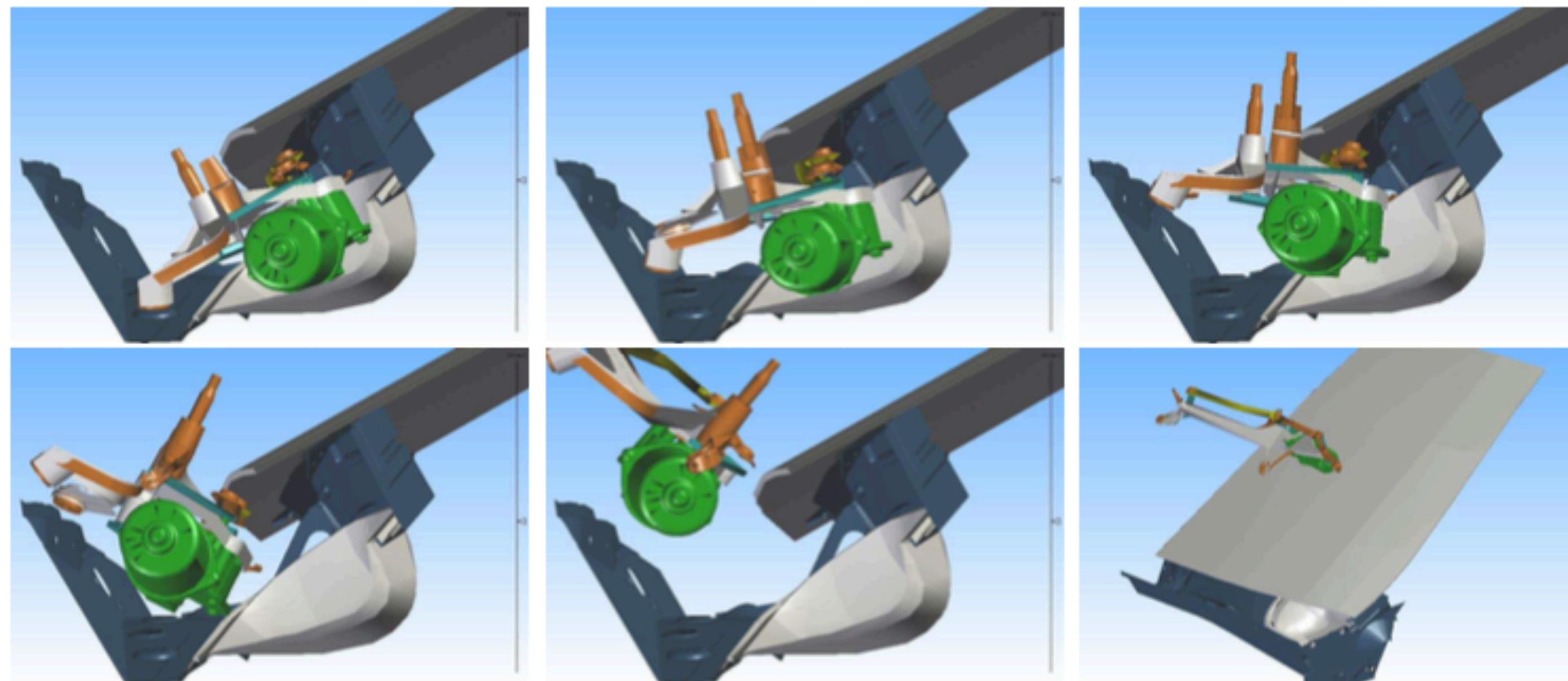


Intelligent and Mobile Robotics Group  
<http://imr.felk.cvut.cz>

Alpha Puzzle 1.0

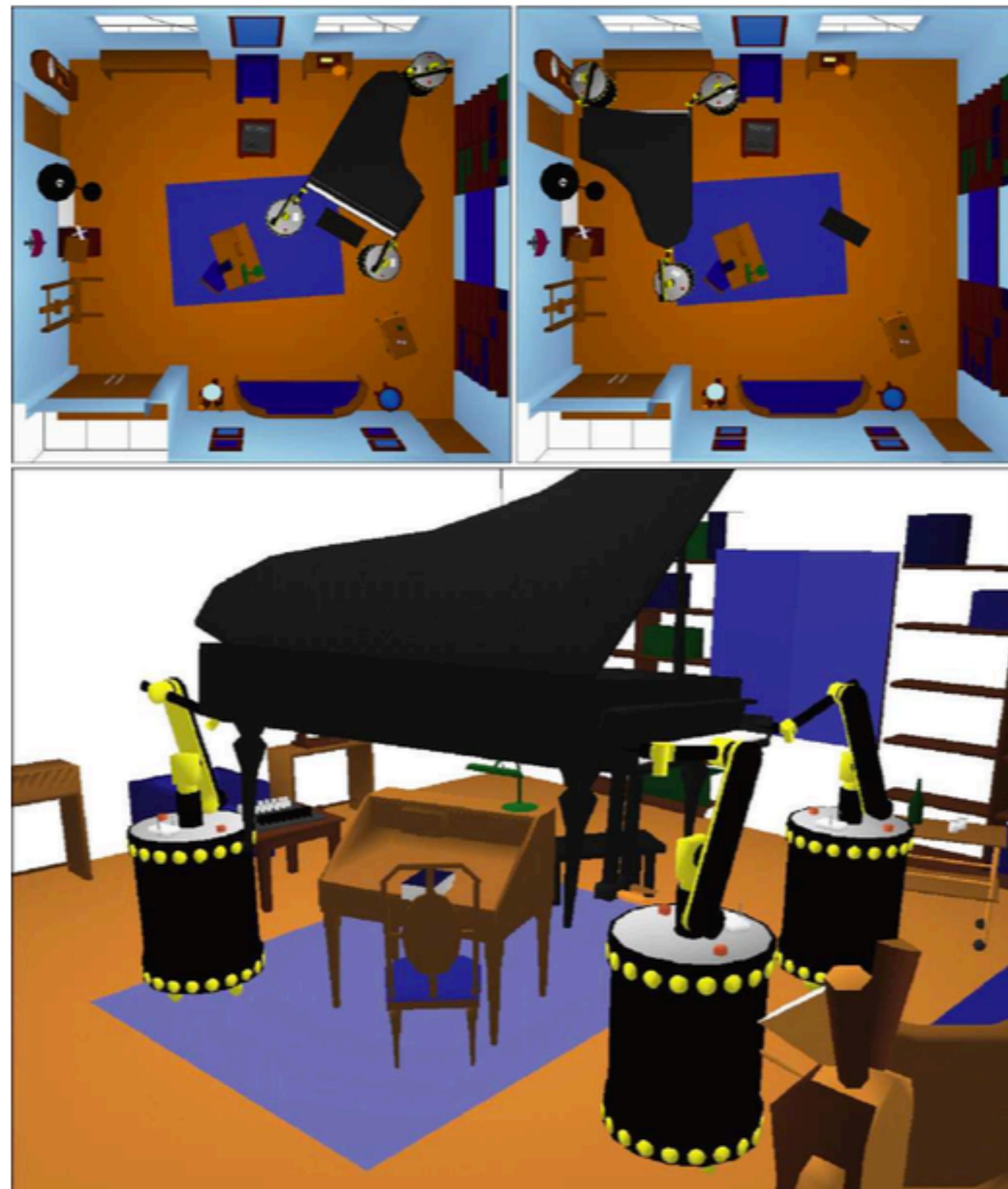


# Automotive Assembly





# Moving Furniture



# Moving Furniture

---

Balancing Exploration and Exploitation in Sampling-Based Motion Planning

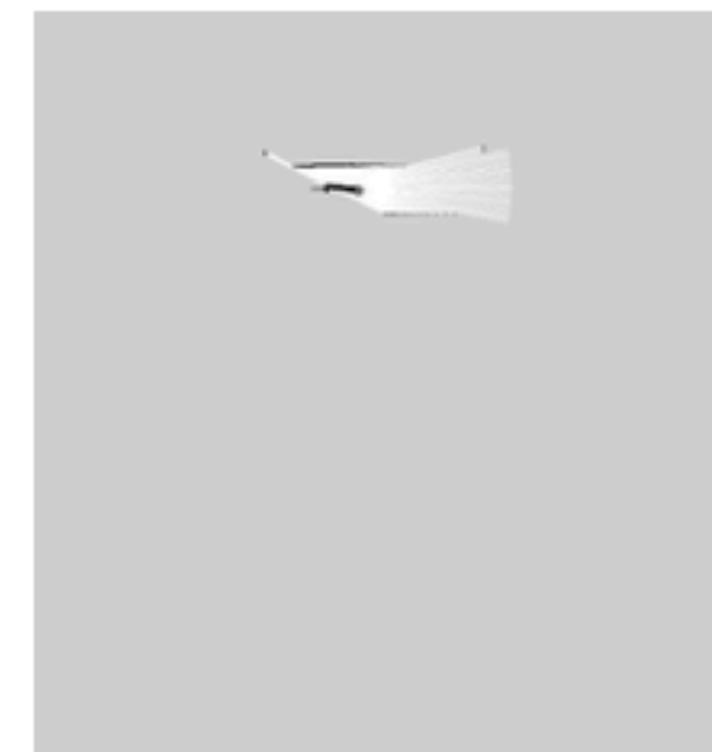
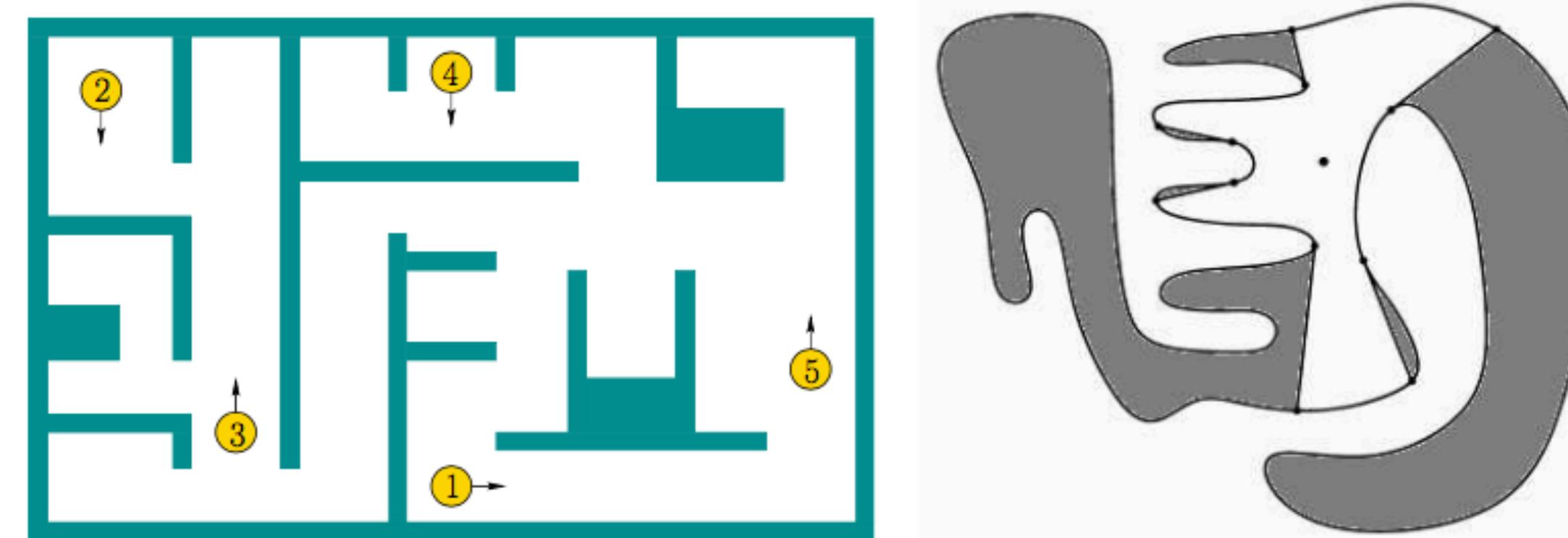
*"The piano mover's problem"*

Markus Rickert, Arne Sieverling, and Oliver Brock

fortiss GmbH, An-Institut Technische Universität München, München, Germany  
Robotics and Biology Laboratory, Technische Universität Berlin, Berlin, Germany

*IEEE Transactions on Robotics*

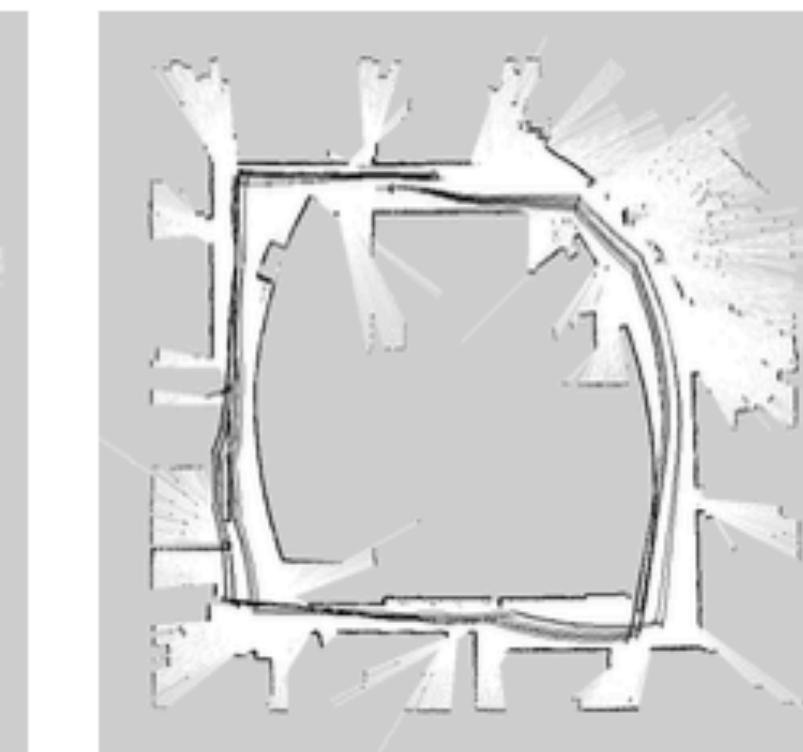
# Navigating Mobile Robots



(a)



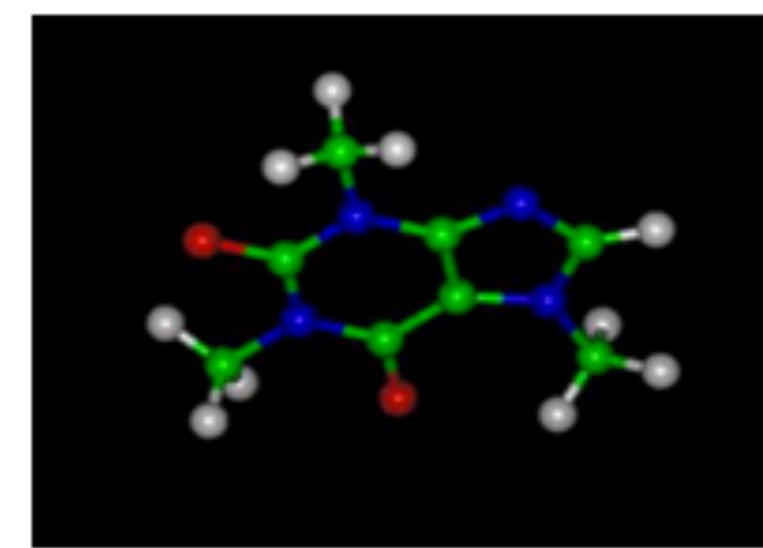
(b)



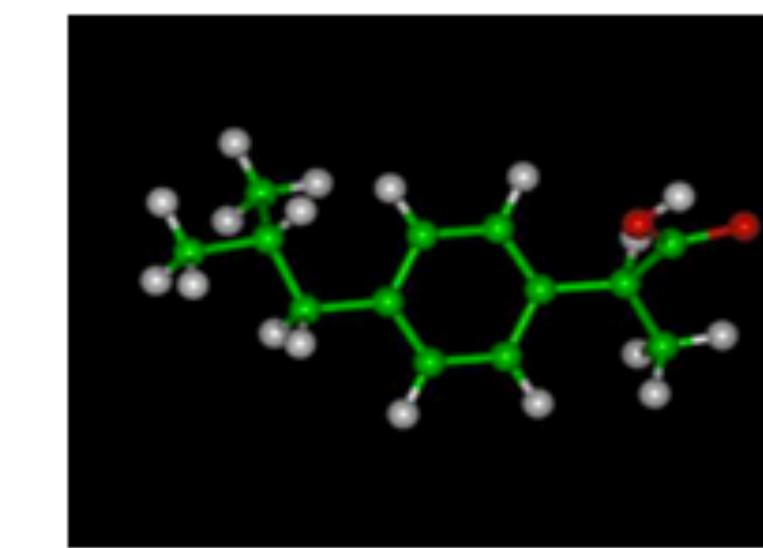
(c)



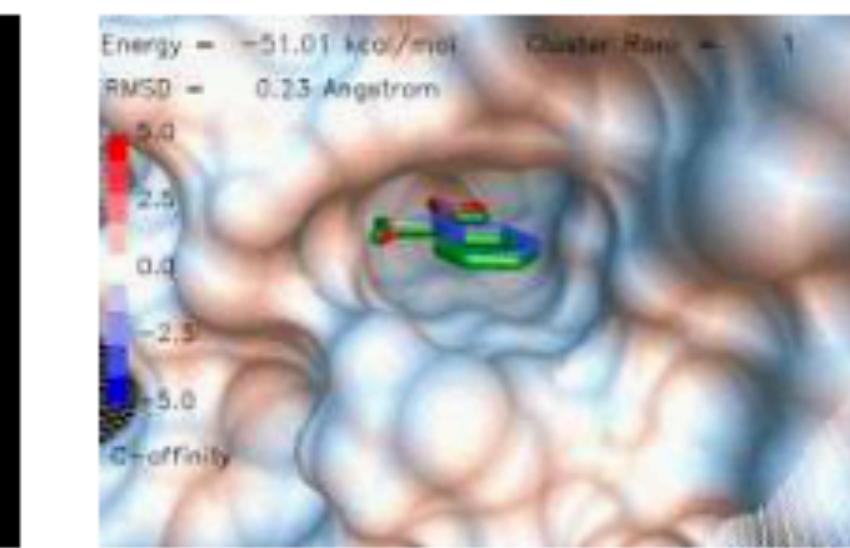
# Computational Biology



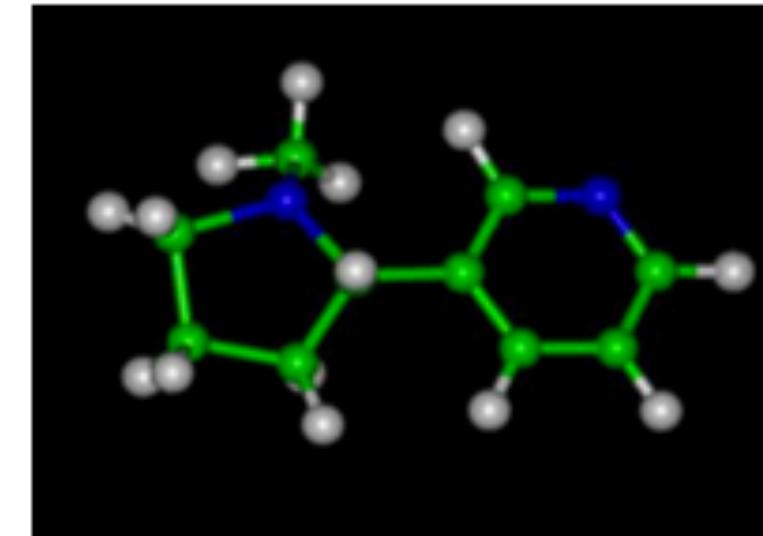
Caffeine



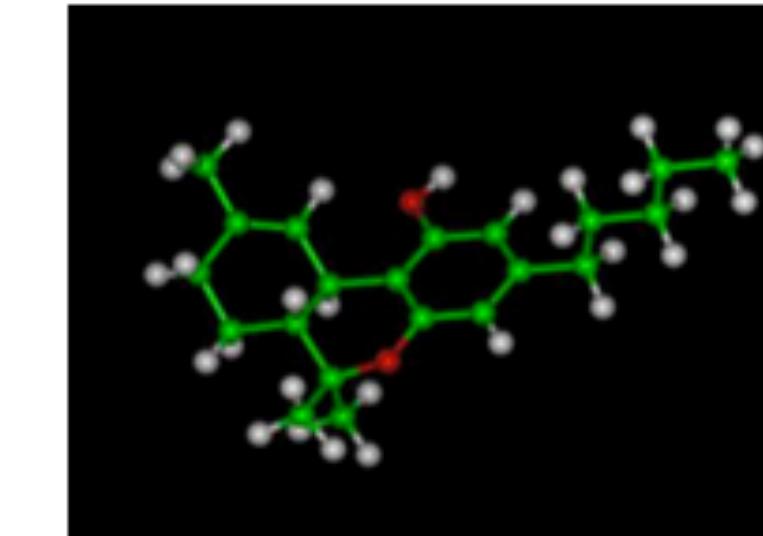
Ibuprofen



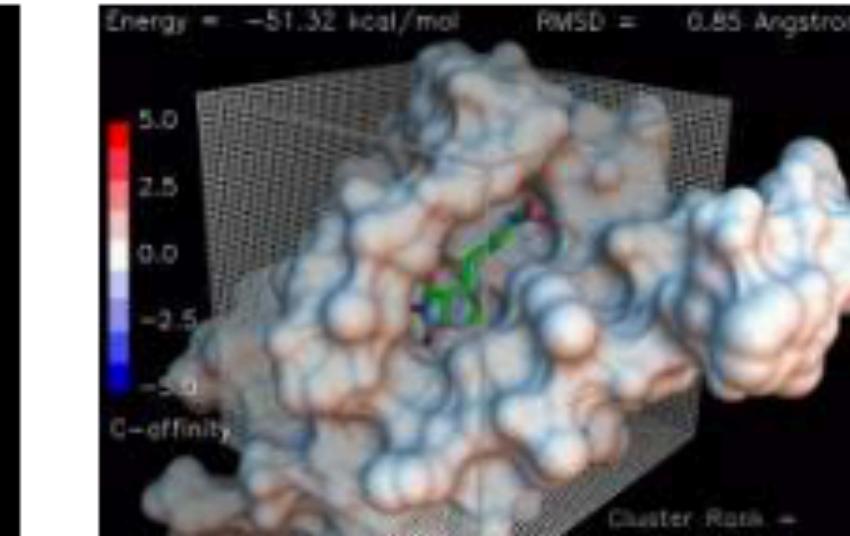
AutoDock



Nicotine



THC



AutoDock

# Ingredients of Planning

**State**

# Ingredients of Planning

**State**

**Time**

# Ingredients of Planning

**State**

**Time**

**Actions**

# Initial and Goal States

## **Criterion**

Feasibility: find a solution

Optimality: find the best solution

# General Forward Search

## **States**

Unvisited  
Dead  
Alive

# General Forward Search

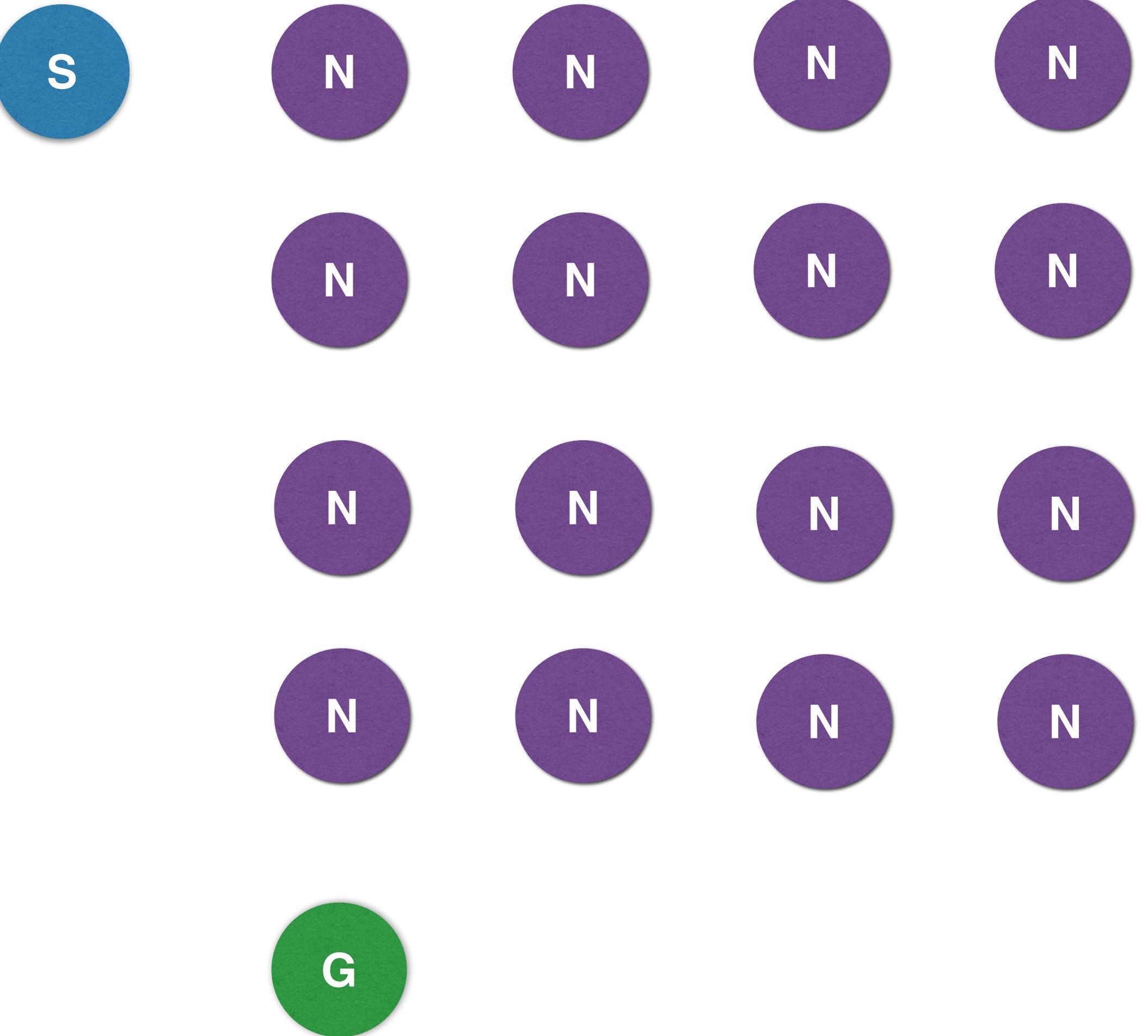
---

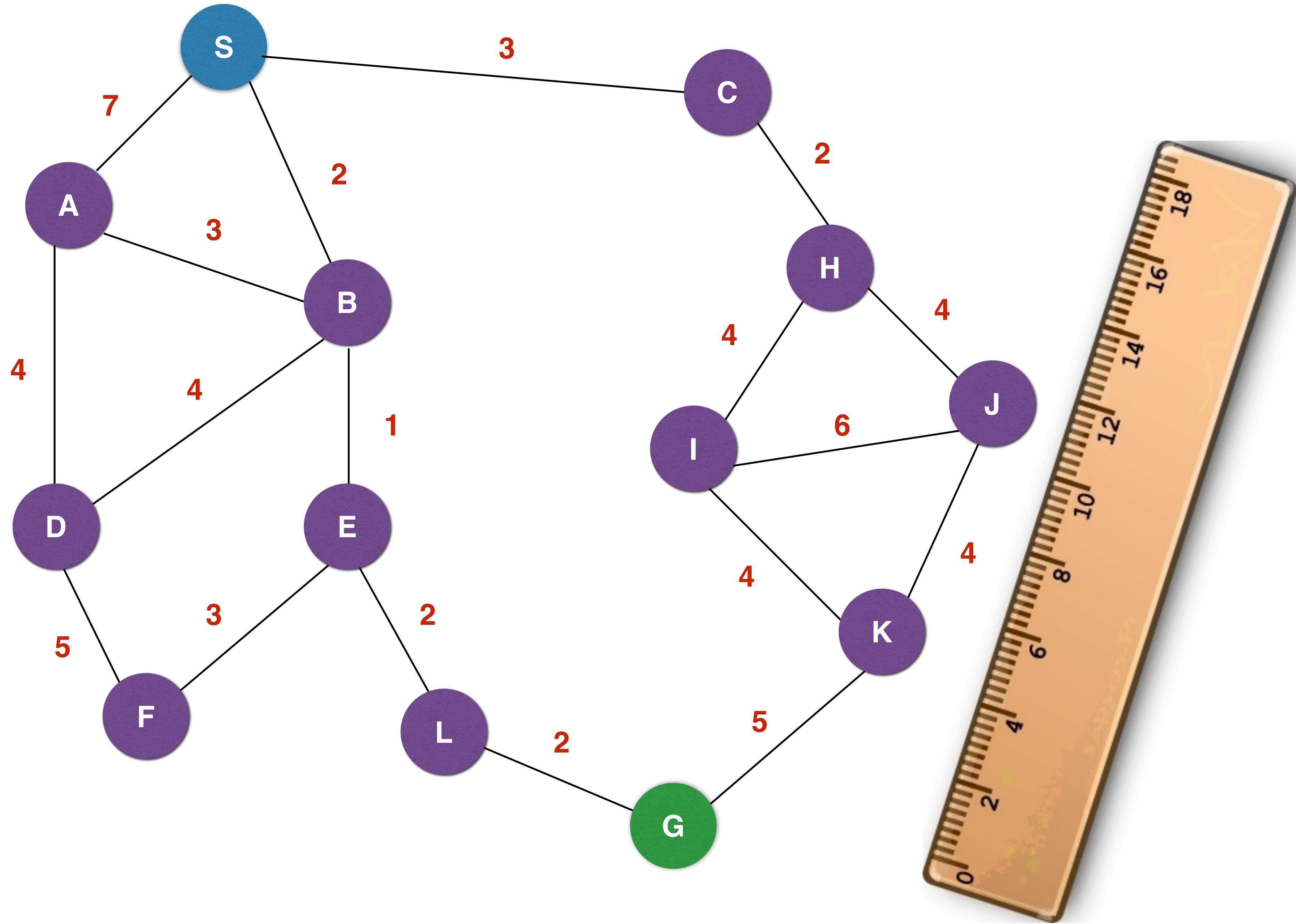
FORWARD\_SEARCH

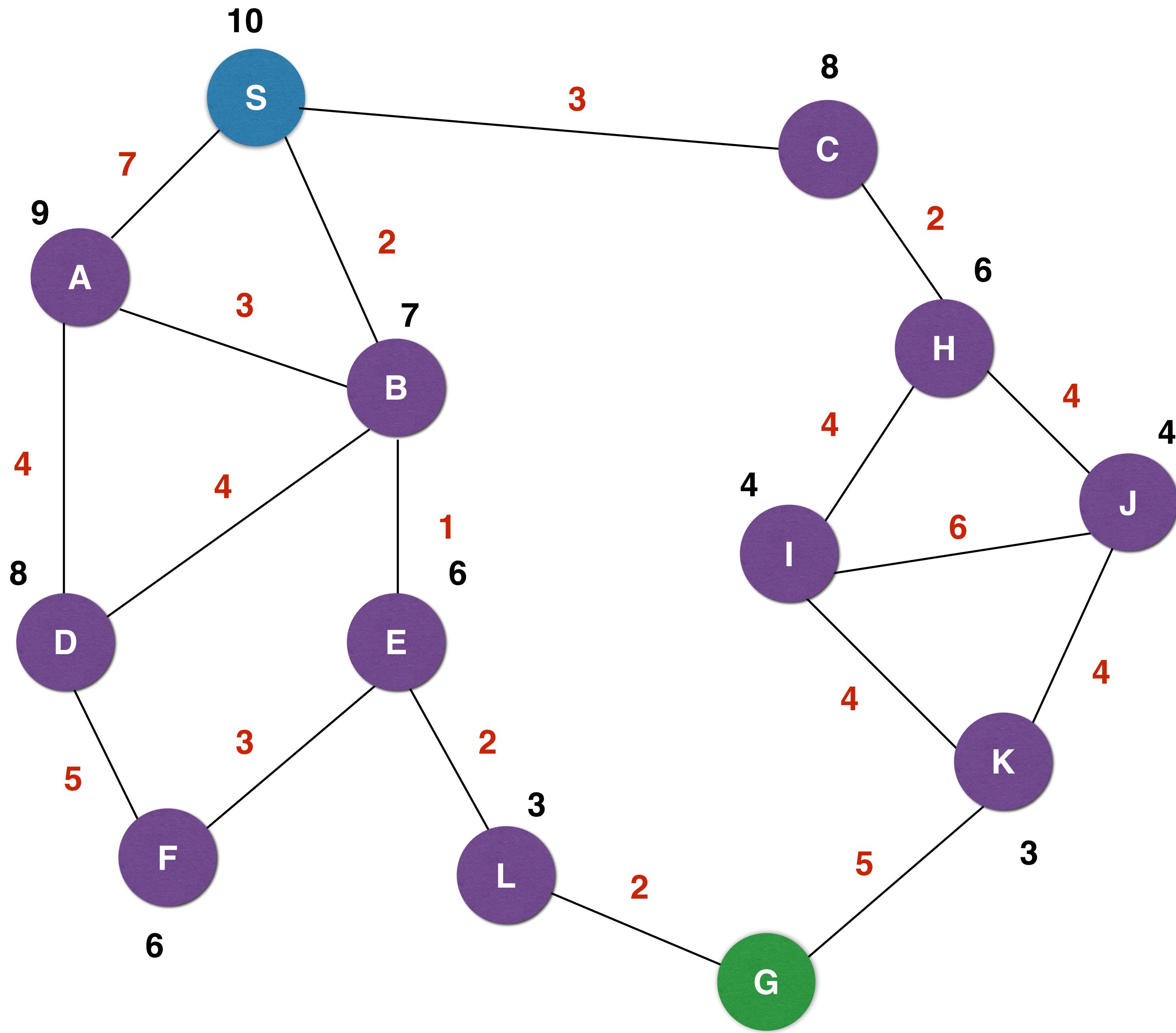
```
1    $Q.Insert(x_I)$  and mark  $x_I$  as visited
2   while  $Q$  not empty do
3        $x \leftarrow Q.GetFirst()$ 
4       if  $x \in X_G$ 
5           return SUCCESS
6       forall  $u \in U(x)$ 
7            $x' \leftarrow f(x, u)$ 
8           if  $x'$  not visited
9               Mark  $x'$  as visited
10               $Q.Insert(x')$ 
11           else
12               Resolve duplicate  $x'$ 
13   return FAILURE
```

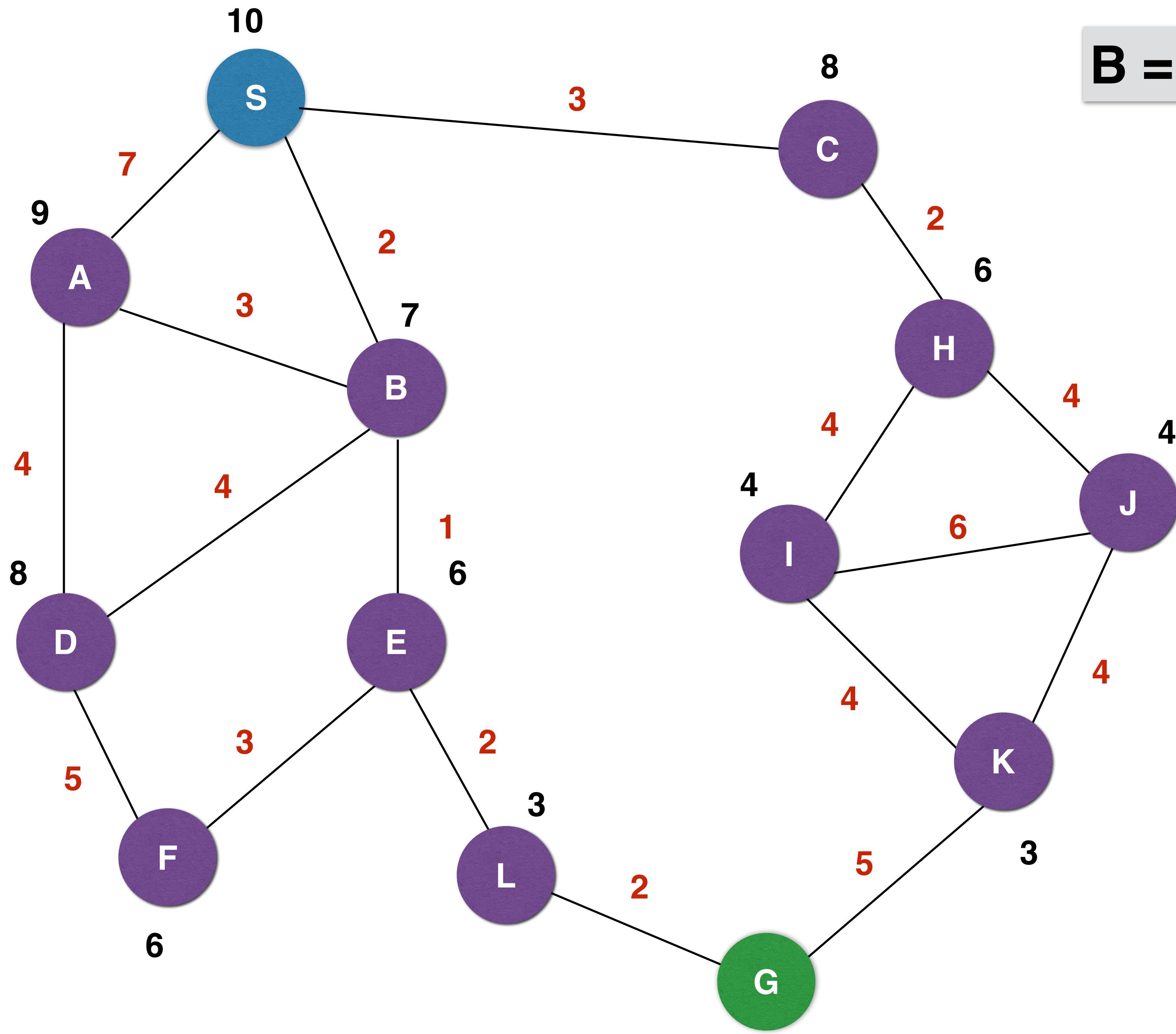
---

# A\* Algorithm

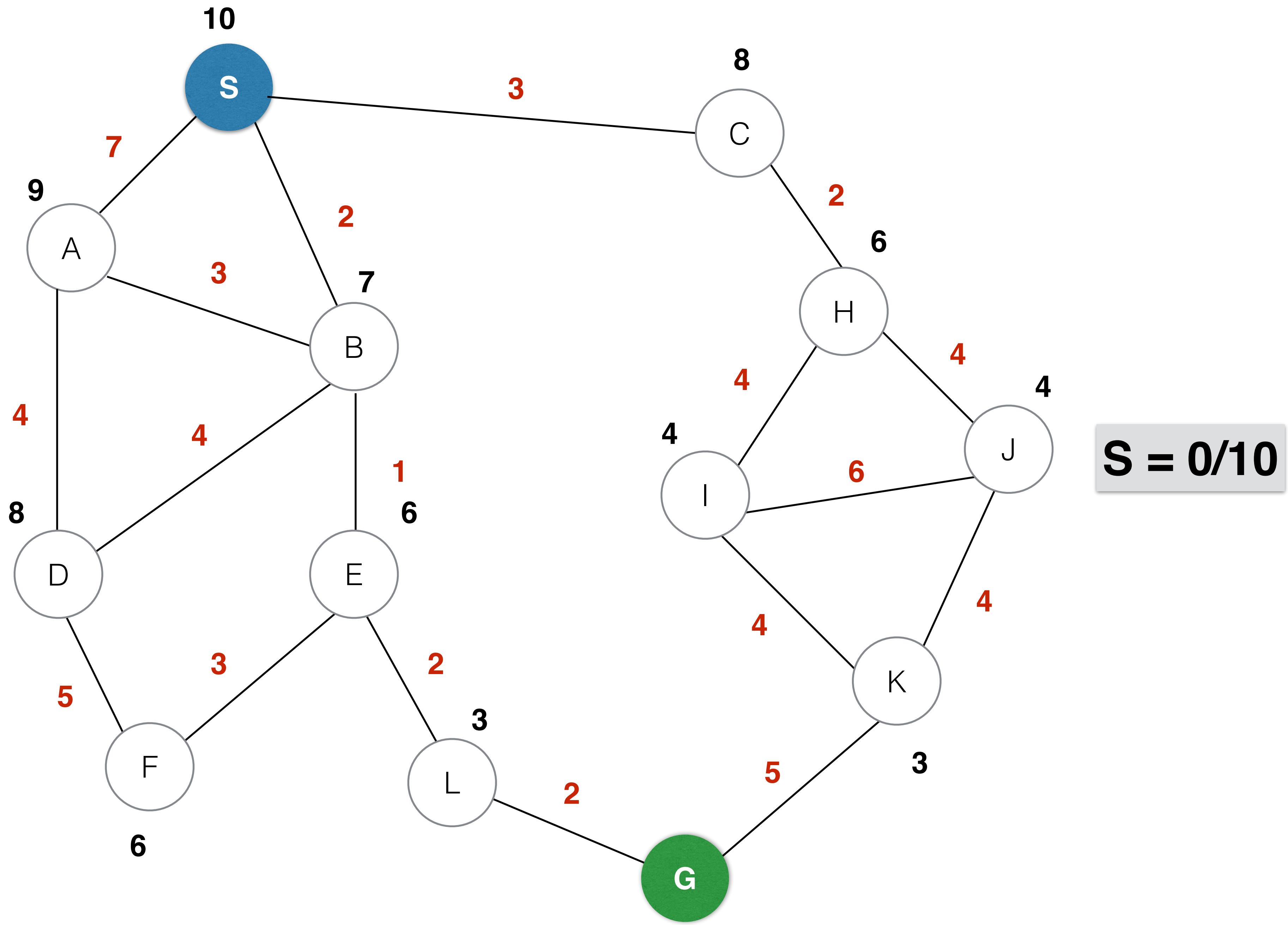


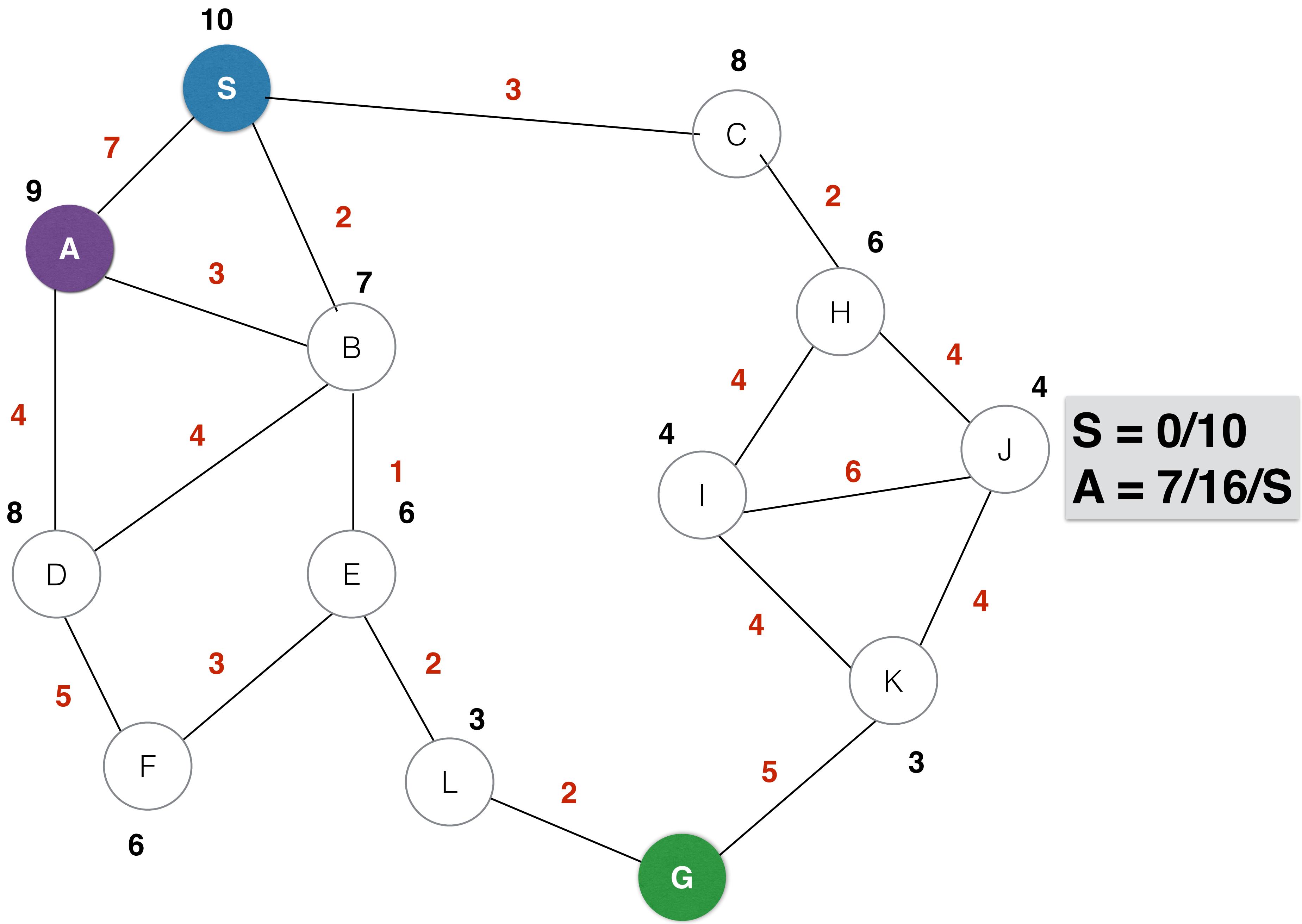


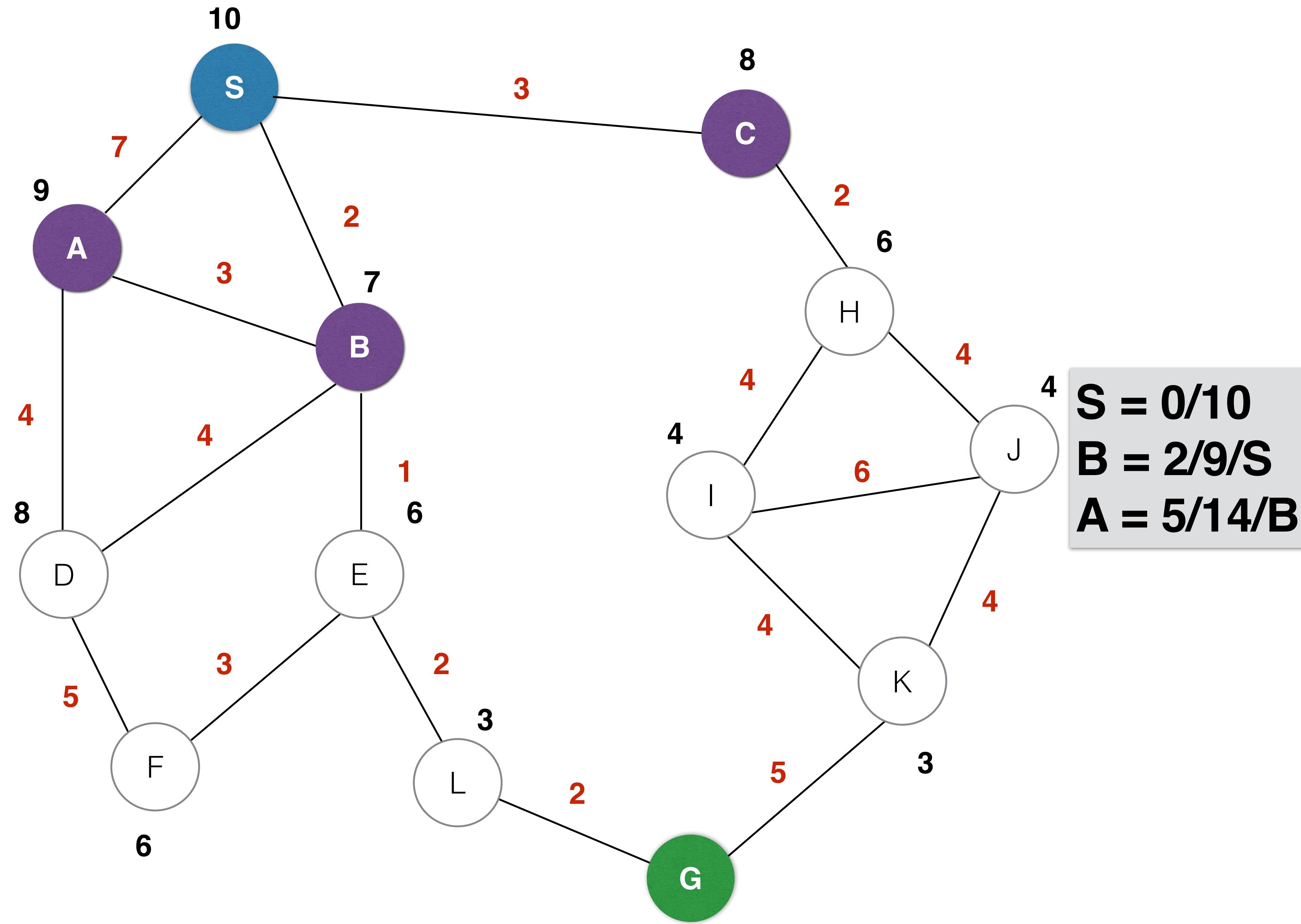


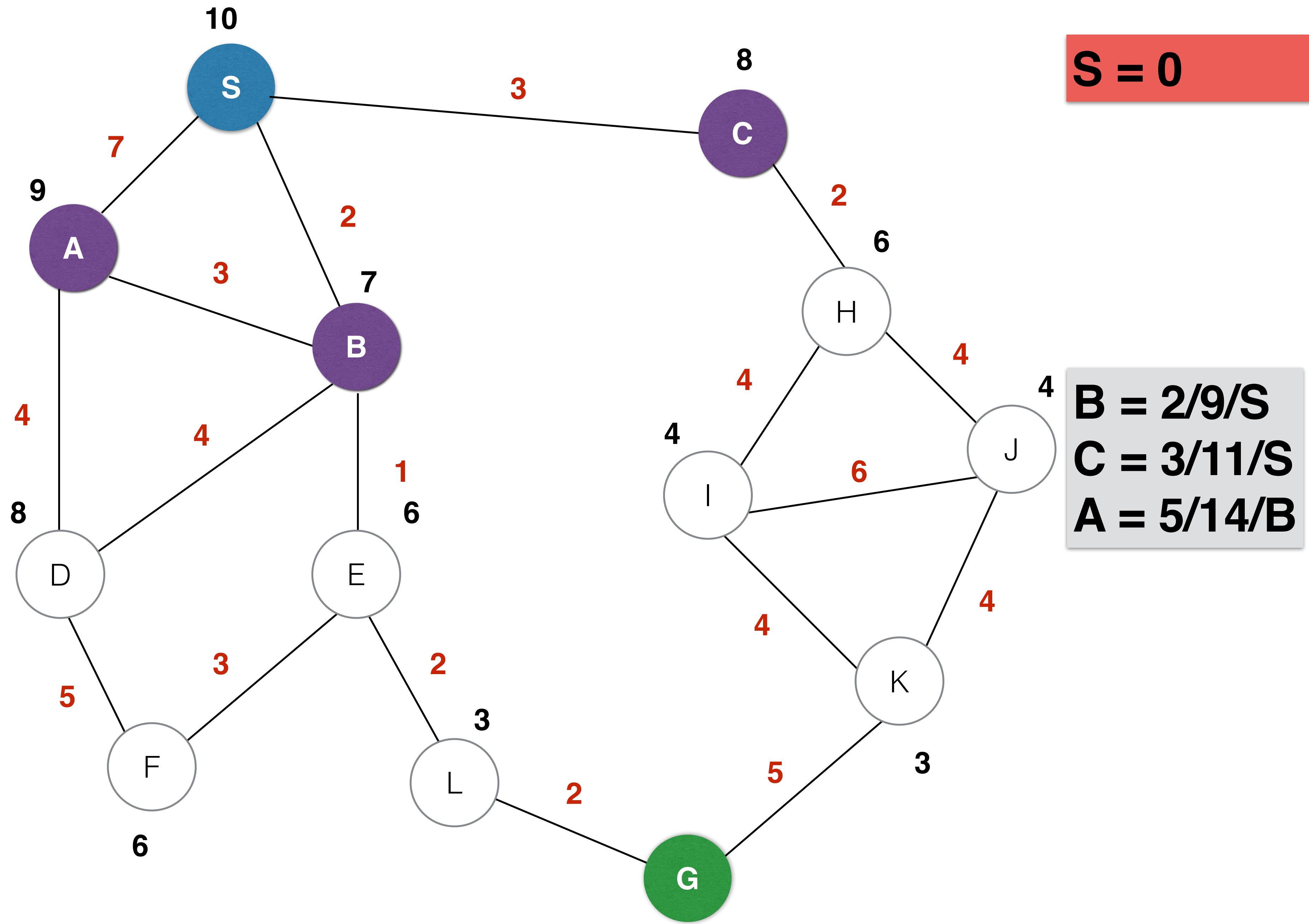


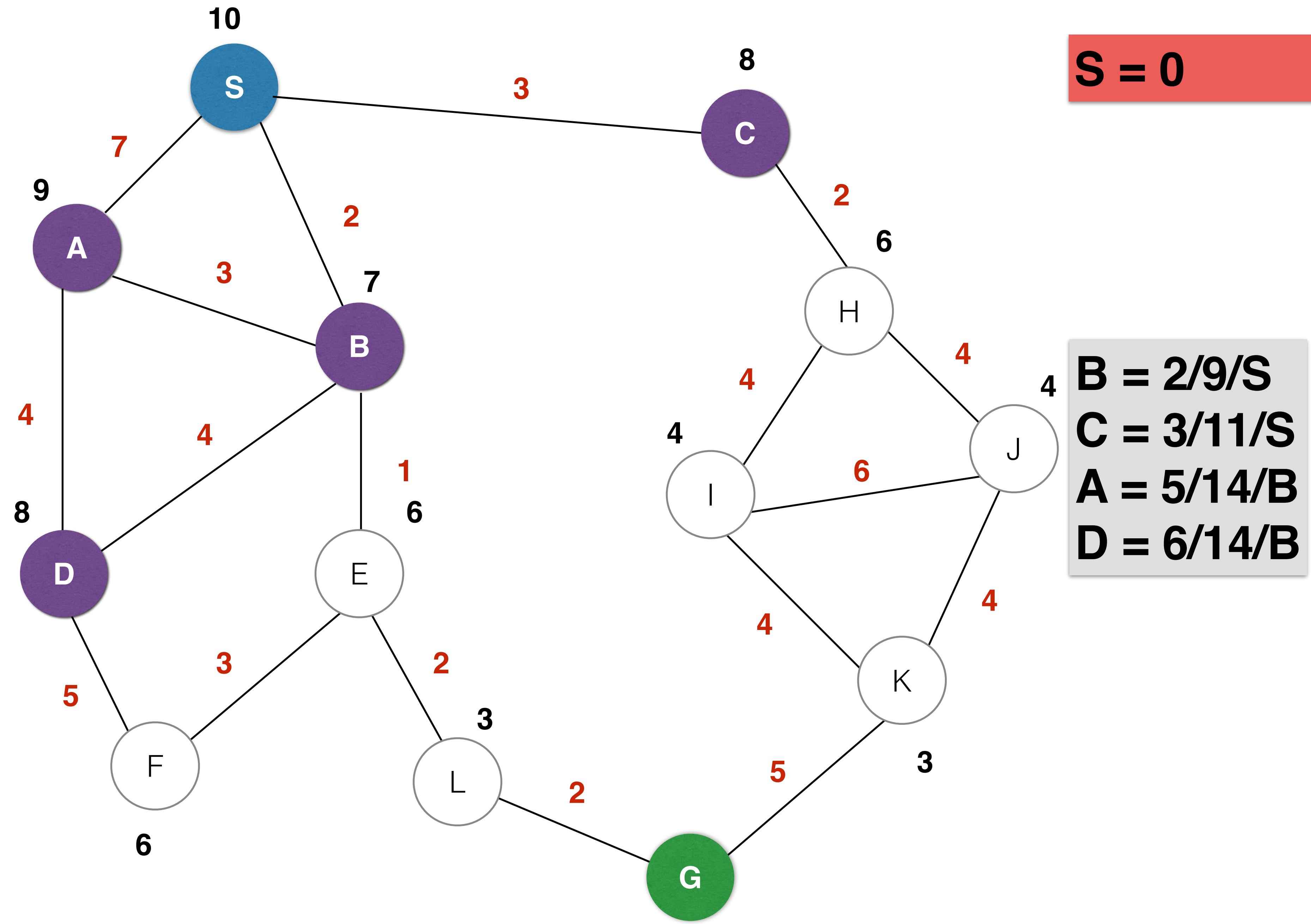
$$B = 2 + 7 = 9$$

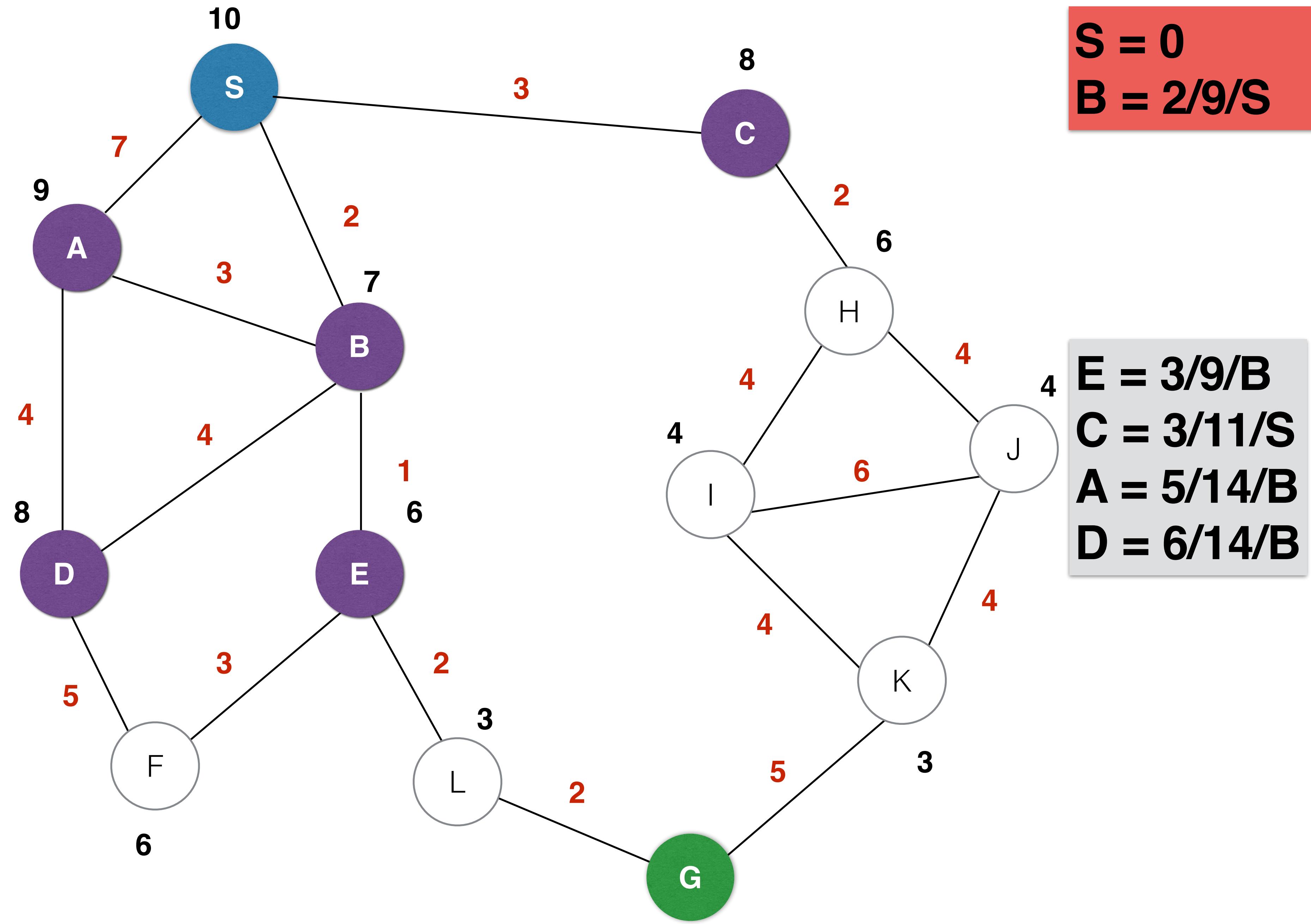


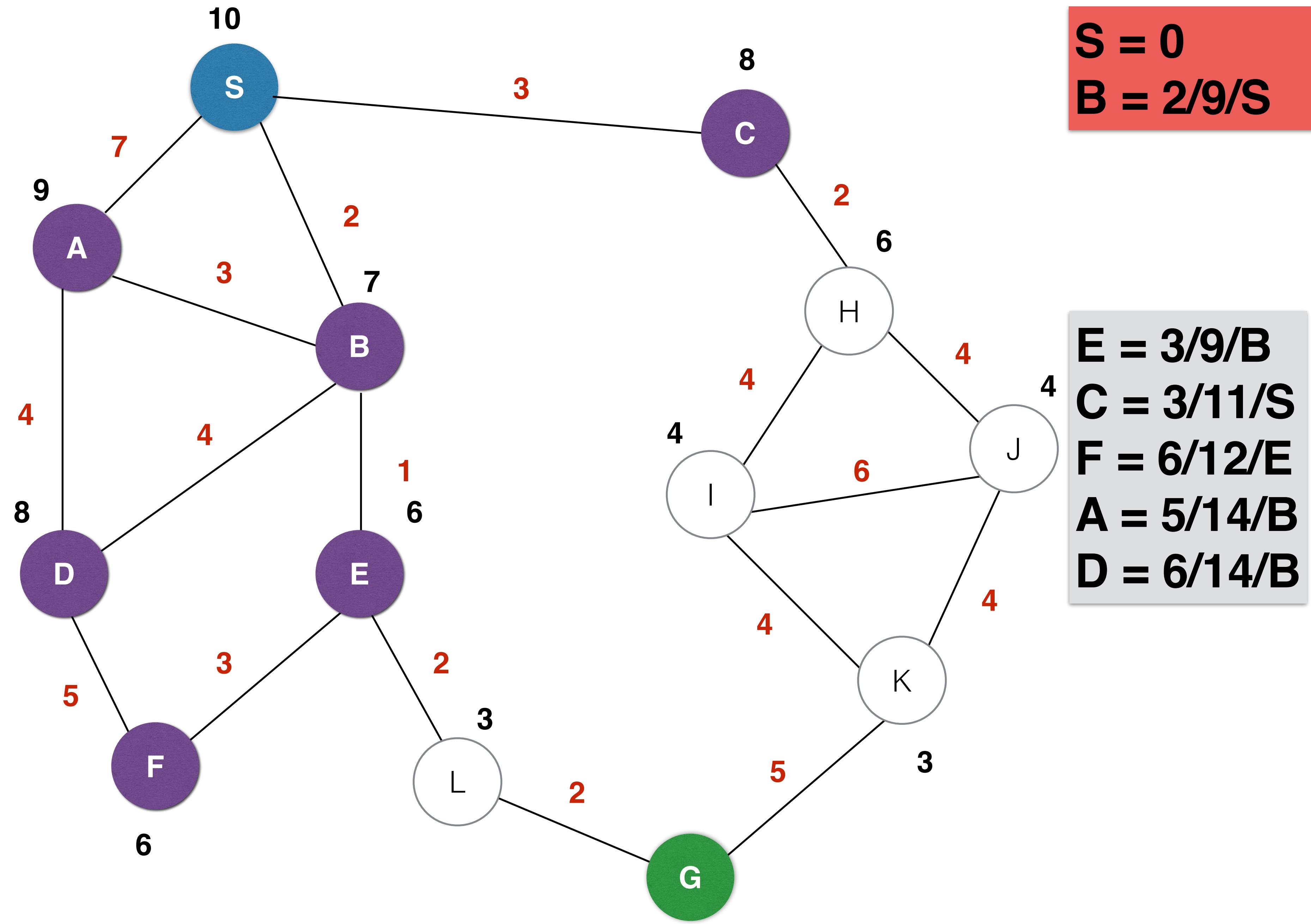


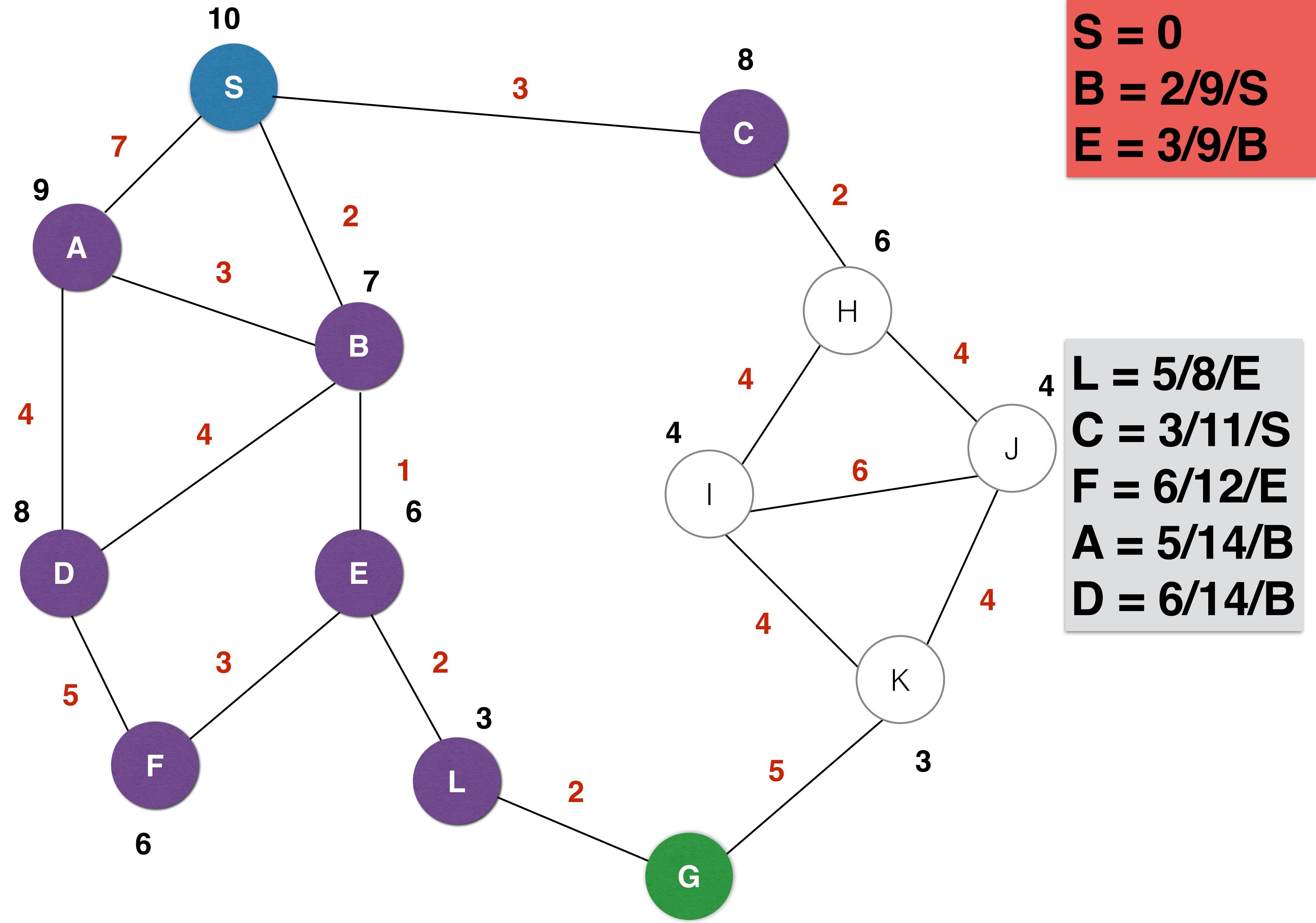


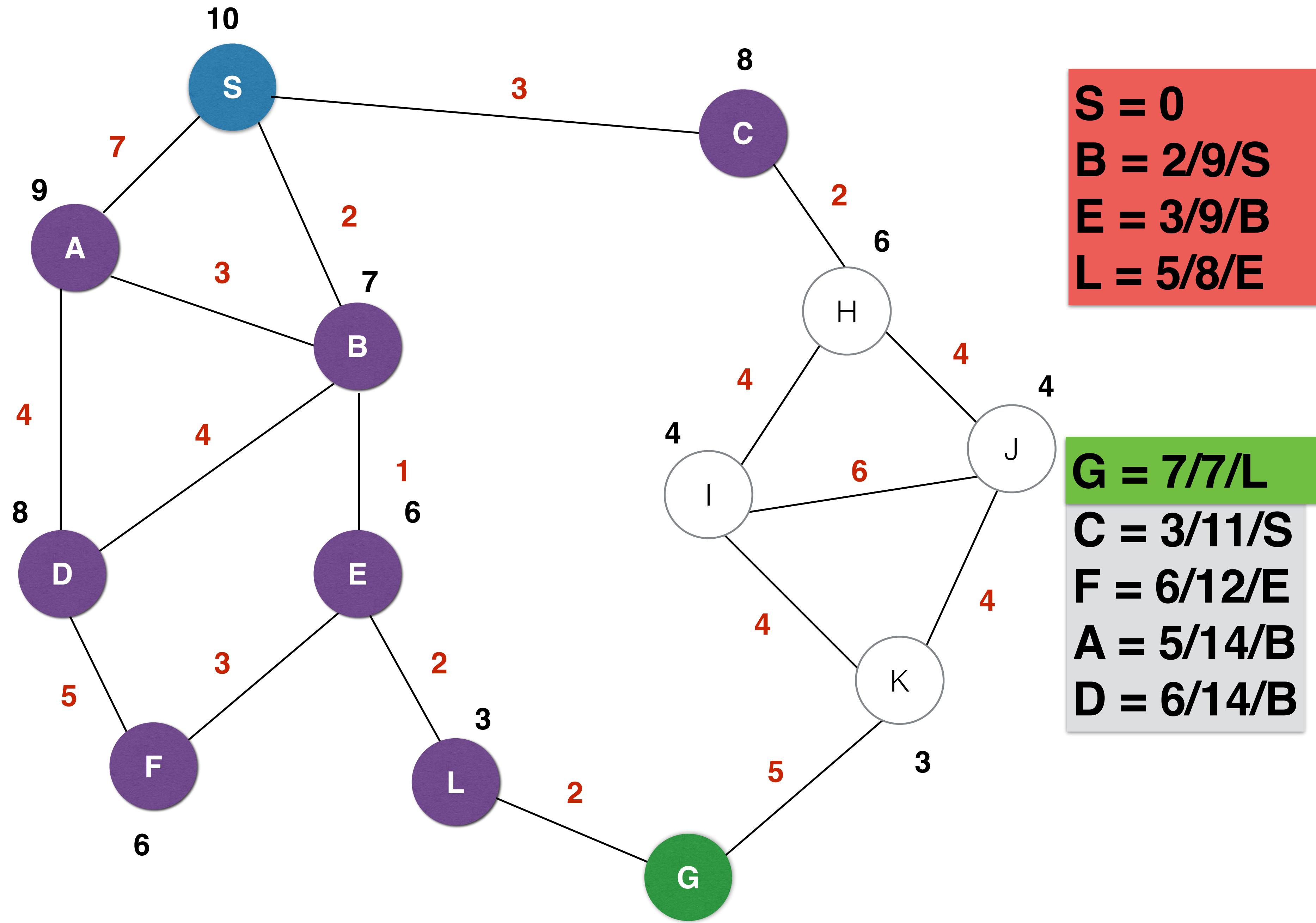


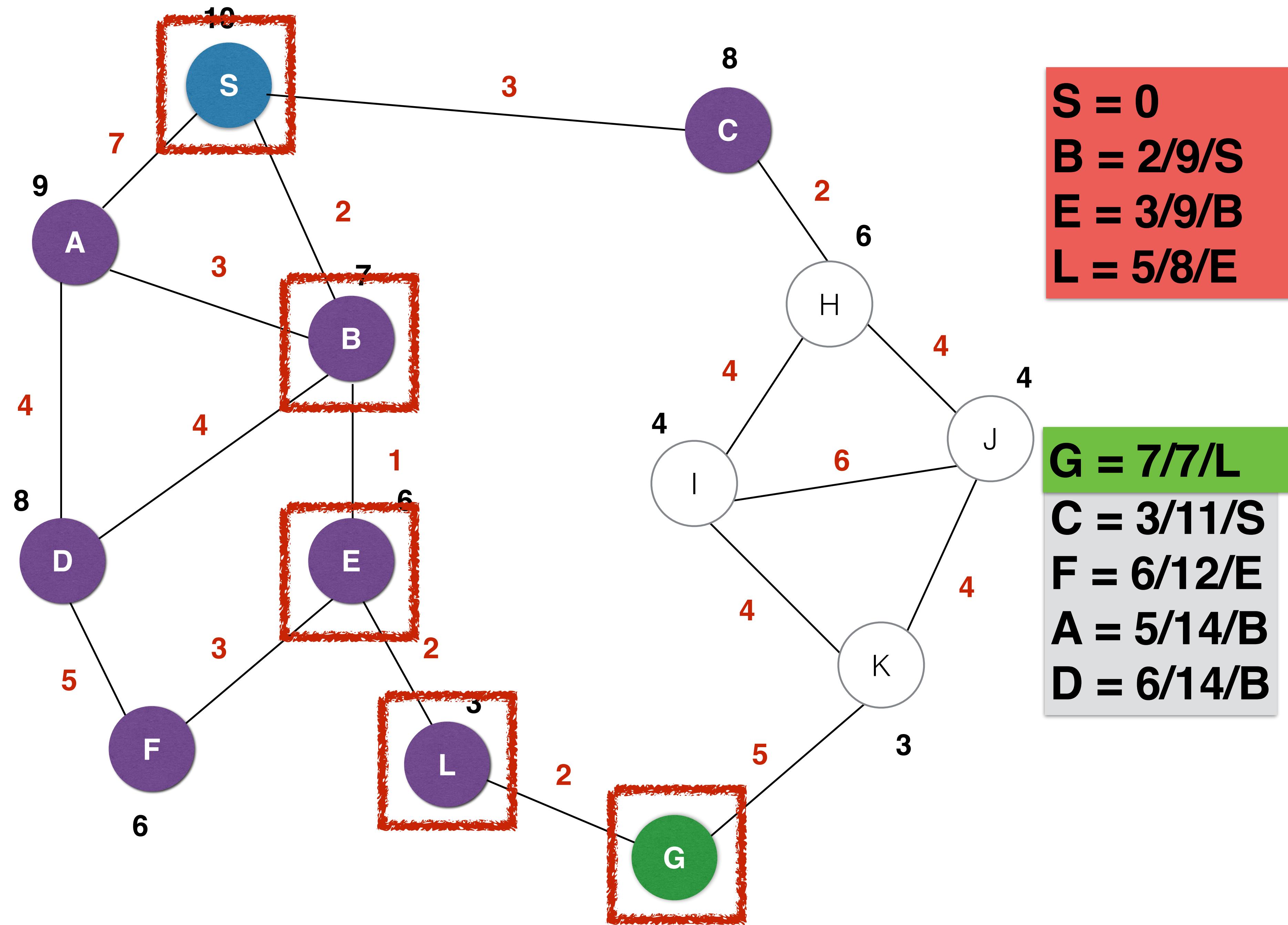












<https://www.youtube.com/watch?v=RYdBcnSiwag>

A\* vs Dijkstra

<https://www.youtube.com/watch?v=xeoGJFwZ714>

