

Graph Breadth First Search

261217 Data Structures for Computer Engineers

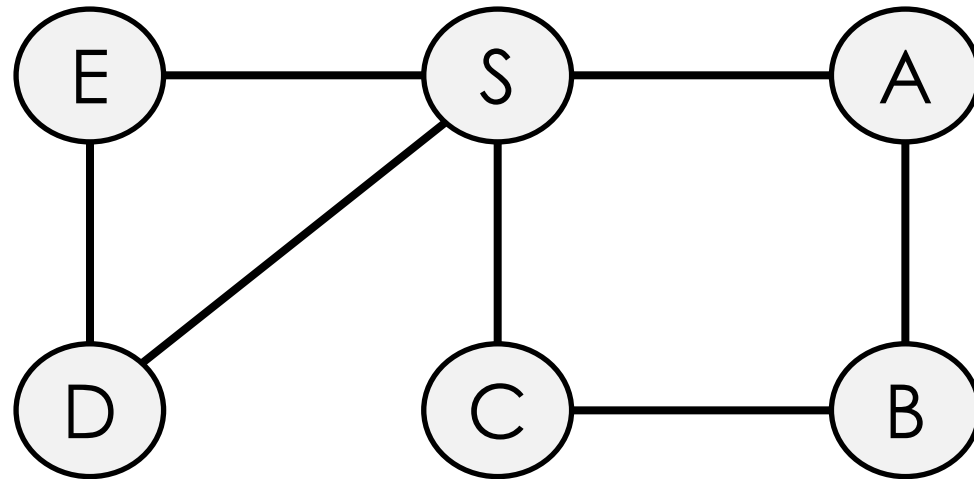
Patiwet Wuttisarnwattana, Ph.D.

patiwet@eng.cmu.ac.th

Computer Engineering, Chiang Mai University

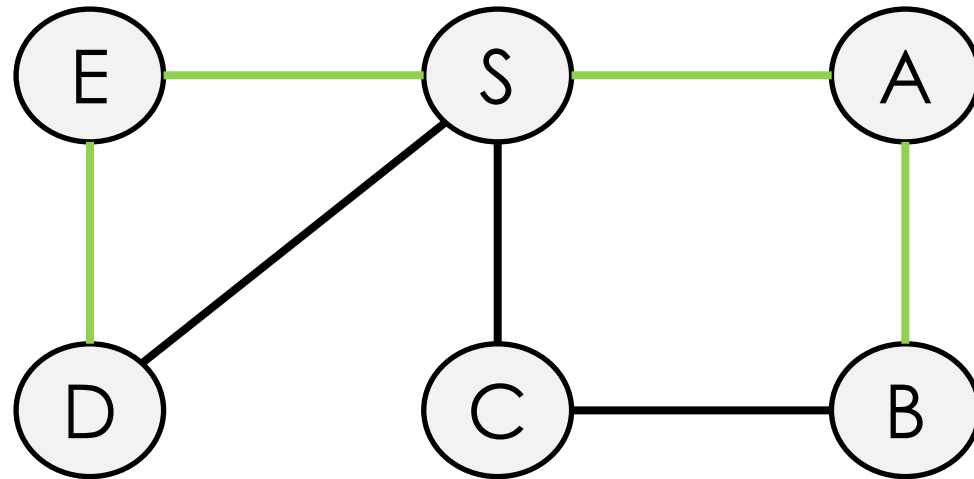
Paths and Lengths

Length of the path $L(P)$ is the number of edges in the path



Paths and Lengths

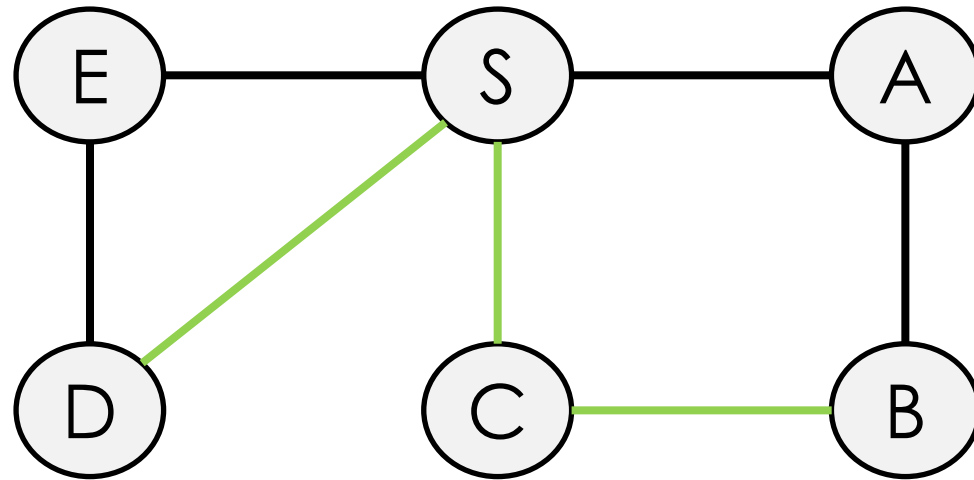
Length of the path $L(P)$ is the number of edges in the path



$$L(D - E - S - A - B) = 4$$

Paths and Lengths

Length of the path $L(P)$ is the number of edges in the path

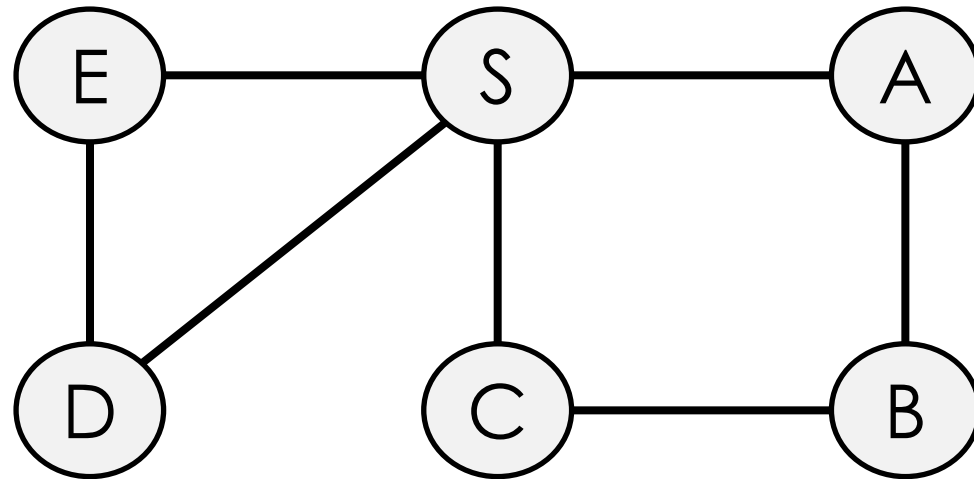


$$L(D - E - S - A - B) = 4$$

$$L(D - S - C - B) = 3$$

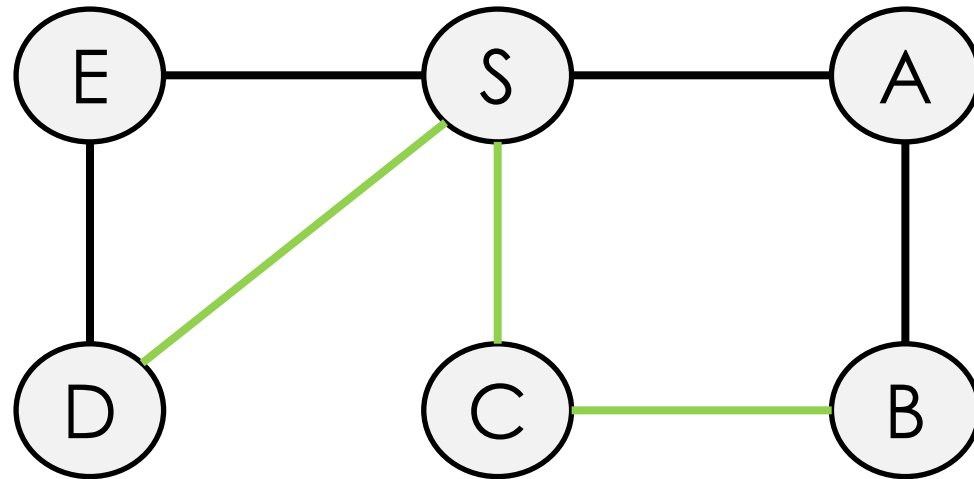
Distances

The **distance** between two vertices is the length of the shortest path between them



Distances

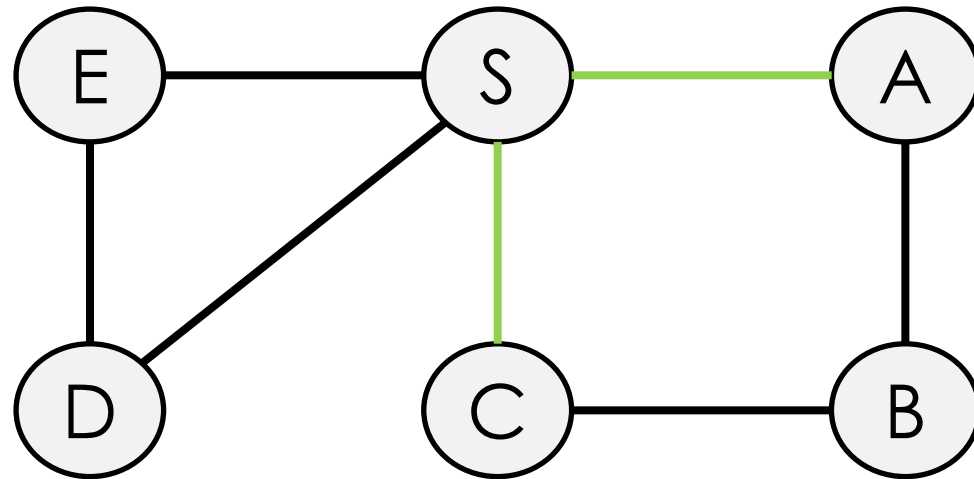
The **distance** between two vertices is the length of the shortest path between them



$$d(D, B) = 3$$

Distances

The **distance** between two vertices is the length of the shortest path between them

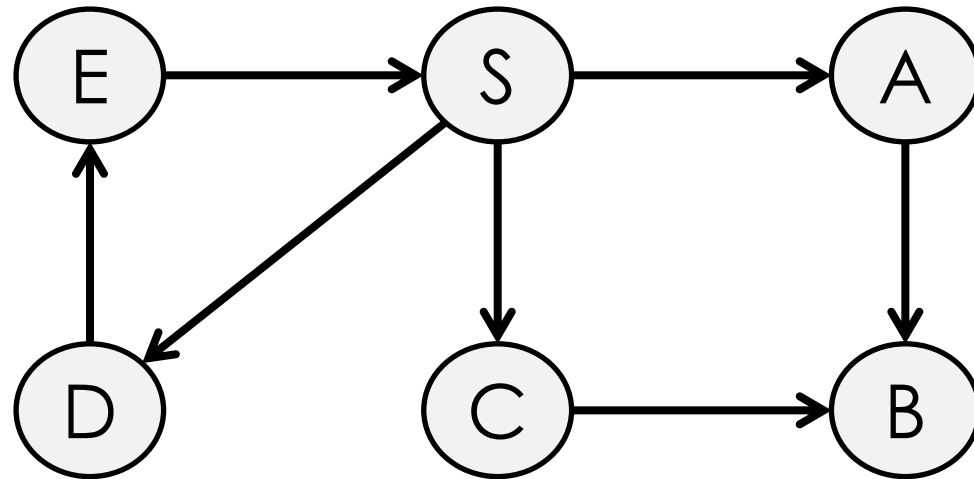


$$d(D, B) = 3$$

$$d(C, A) = 2$$

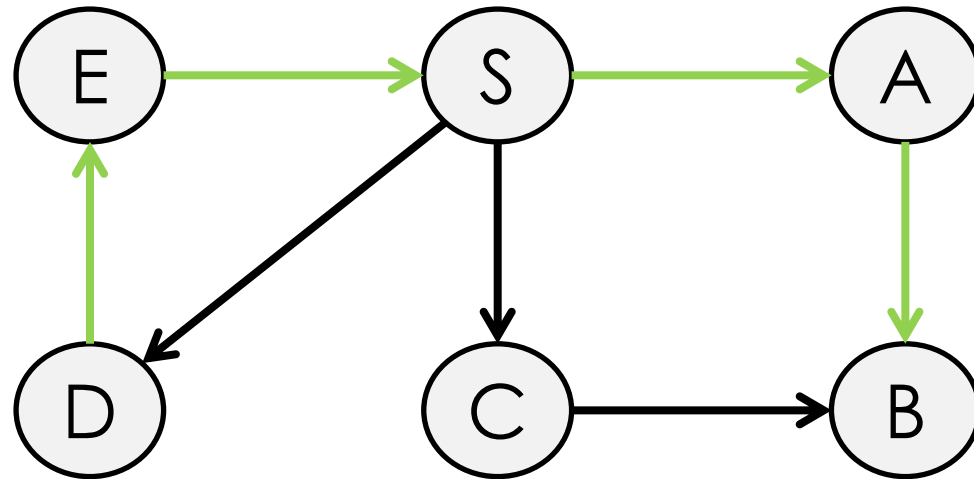
Distances

The **distance** between two vertices is the length of the shortest path between them



Distances

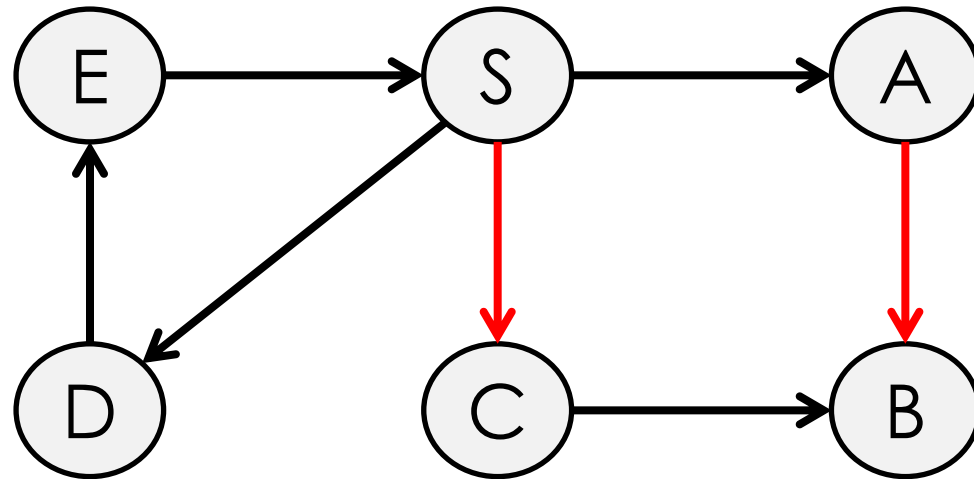
The **distance** between two vertices is the length of the shortest path between them



$$d(D, B) = 4$$

Distances

The **distance** between two vertices is the length of the shortest path between them

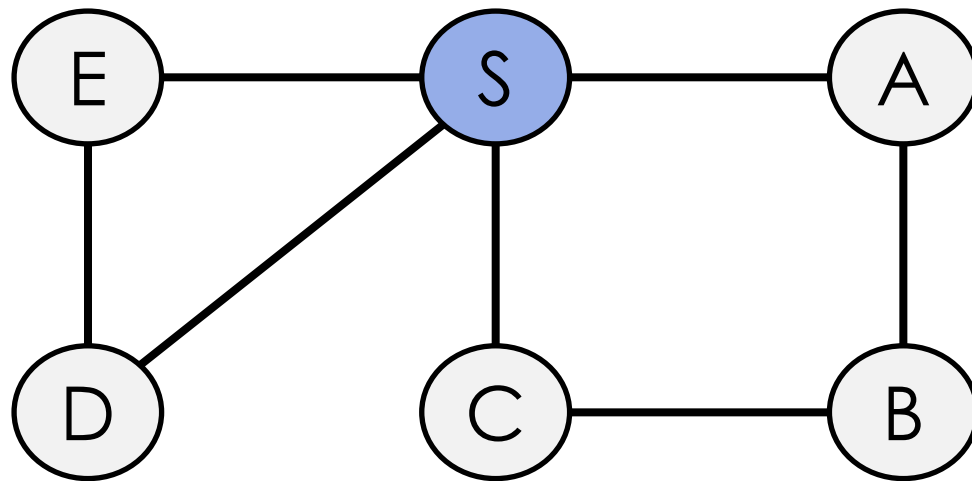


$$d(D, B) = 4$$

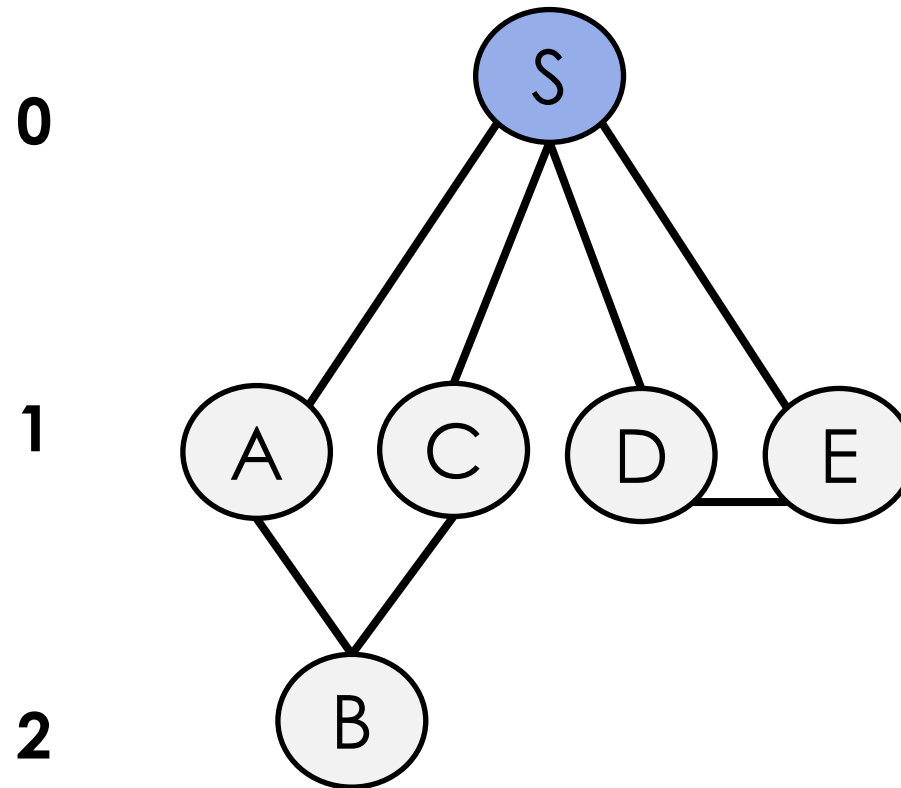
$$d(C, A) = \infty$$

Distances

Going from S to other vertices

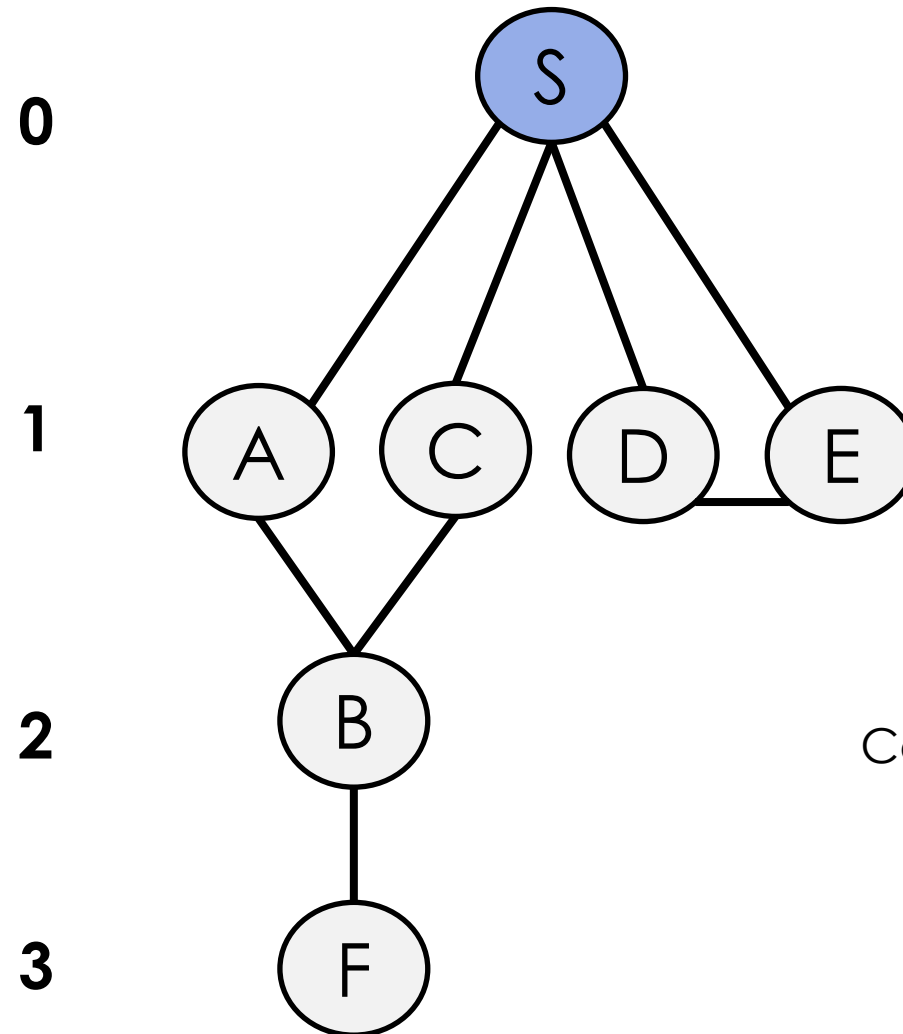


Distance Layers



Distance from vertex S to other vertices

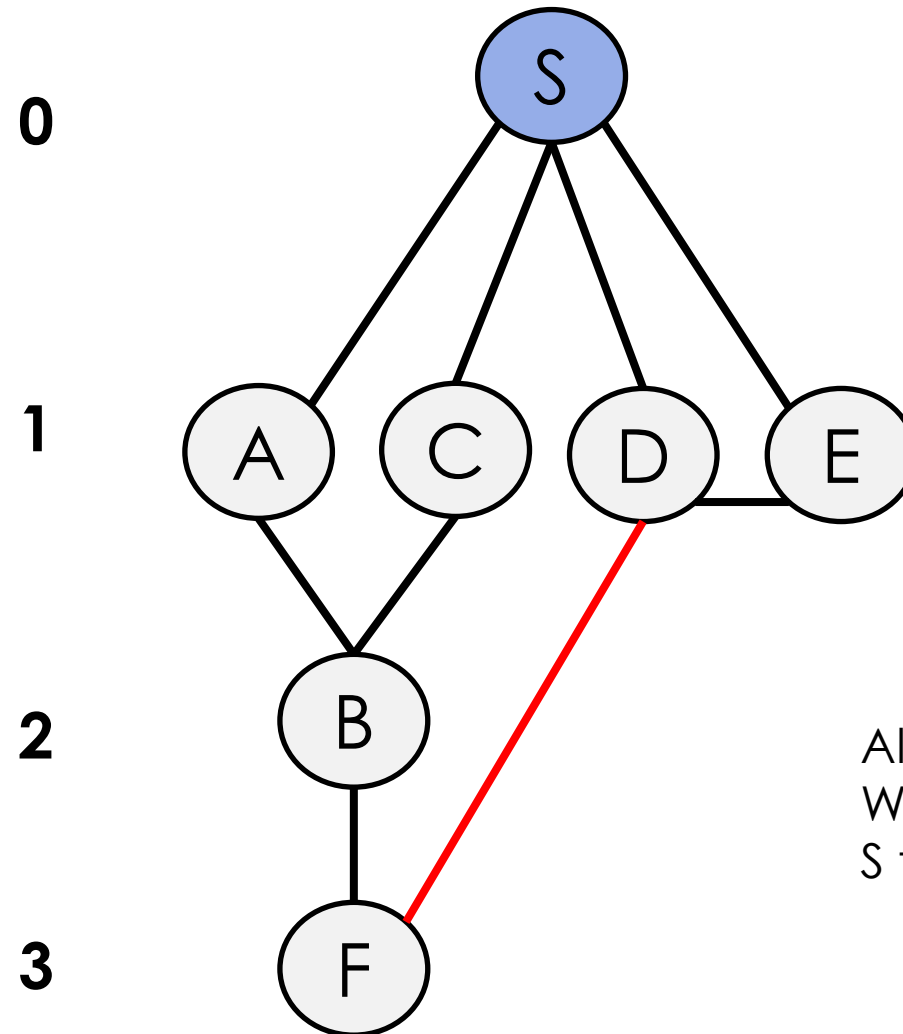
Distance Layers



Connect vertex F to B?

Distance from vertex S to other vertices

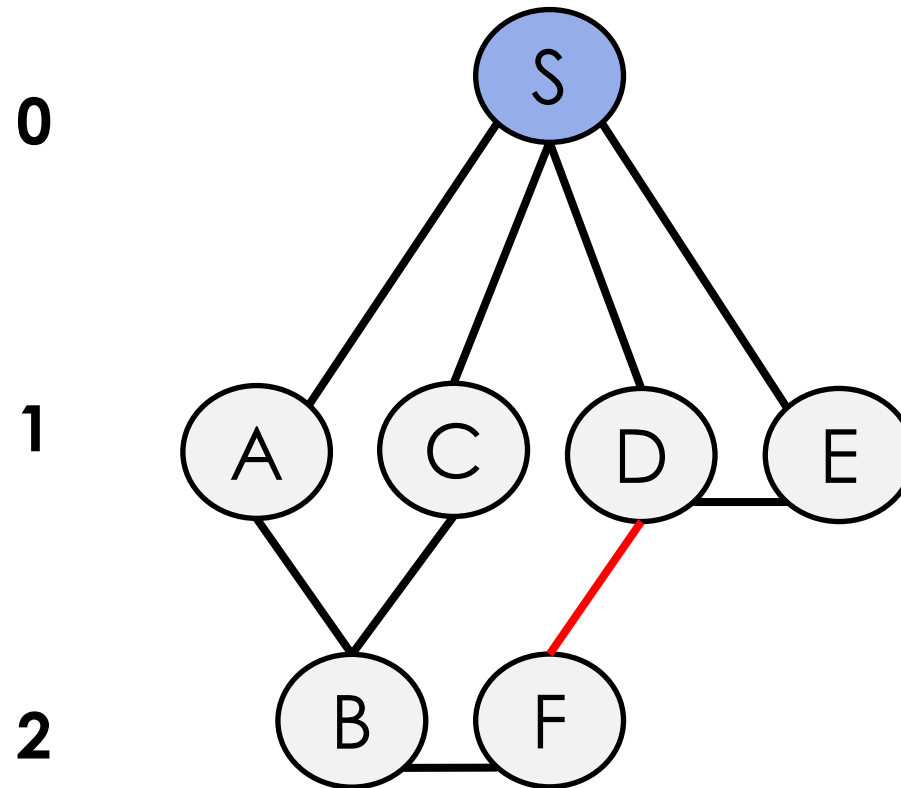
Distance Layers



Also connect F to E
What is the distance from
S to F?

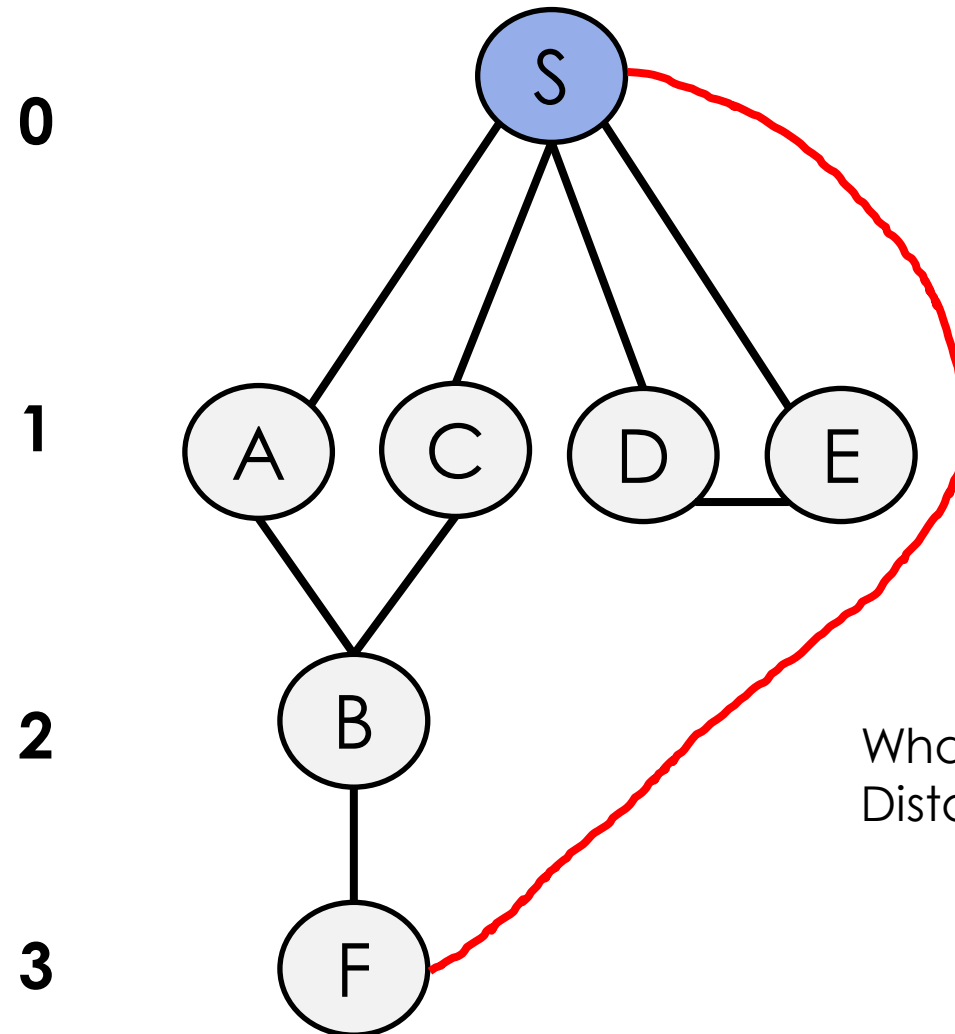
Distance from vertex S to other vertices

Distance Layers



Distance from vertex S to other vertices

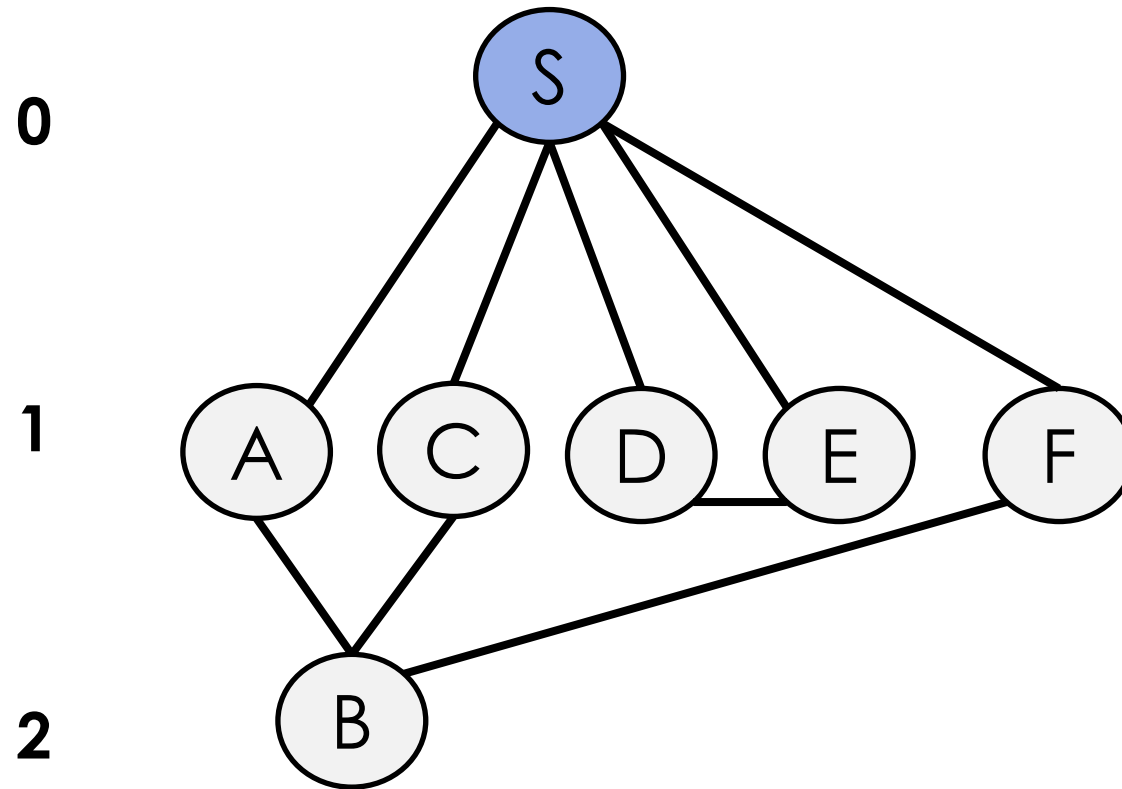
Distance Layers



What if you connect F to S
Distance of F to S?

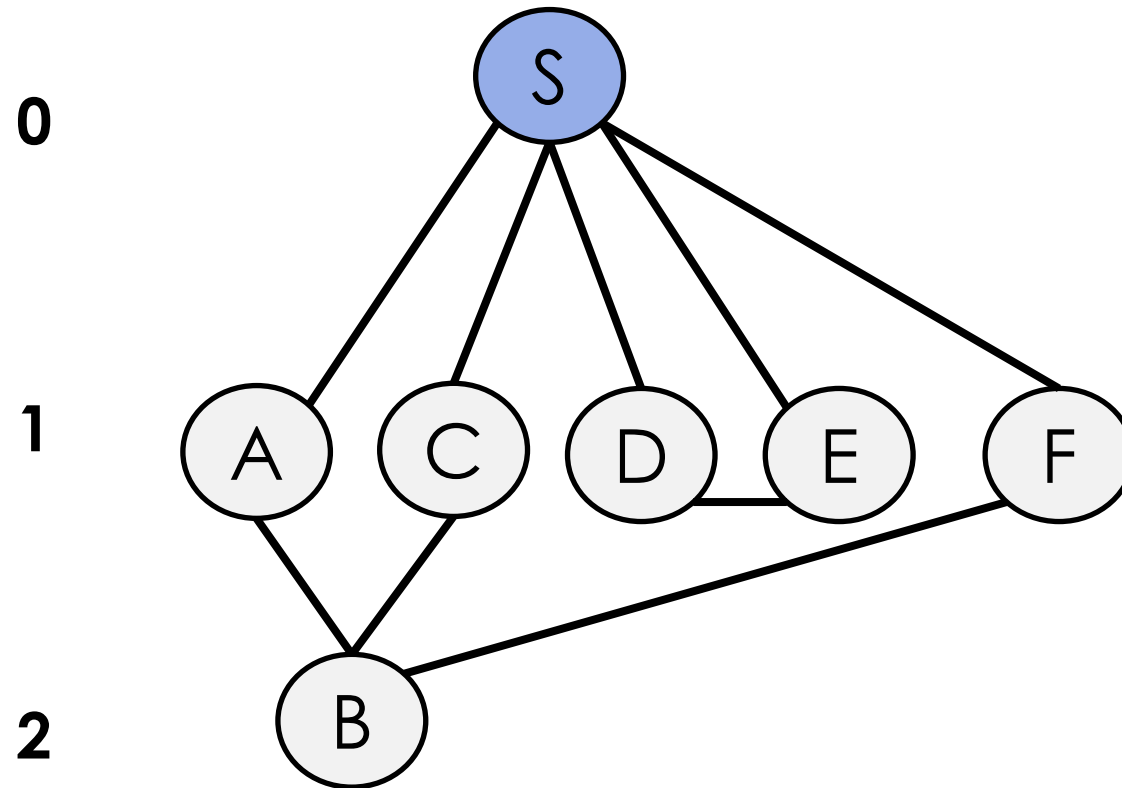
Distance from vertex S to other vertices

Distance Layers



Distance from vertex S to other vertices

Distance Layers



Distance from vertex S to other vertices

Breadth First Traversal

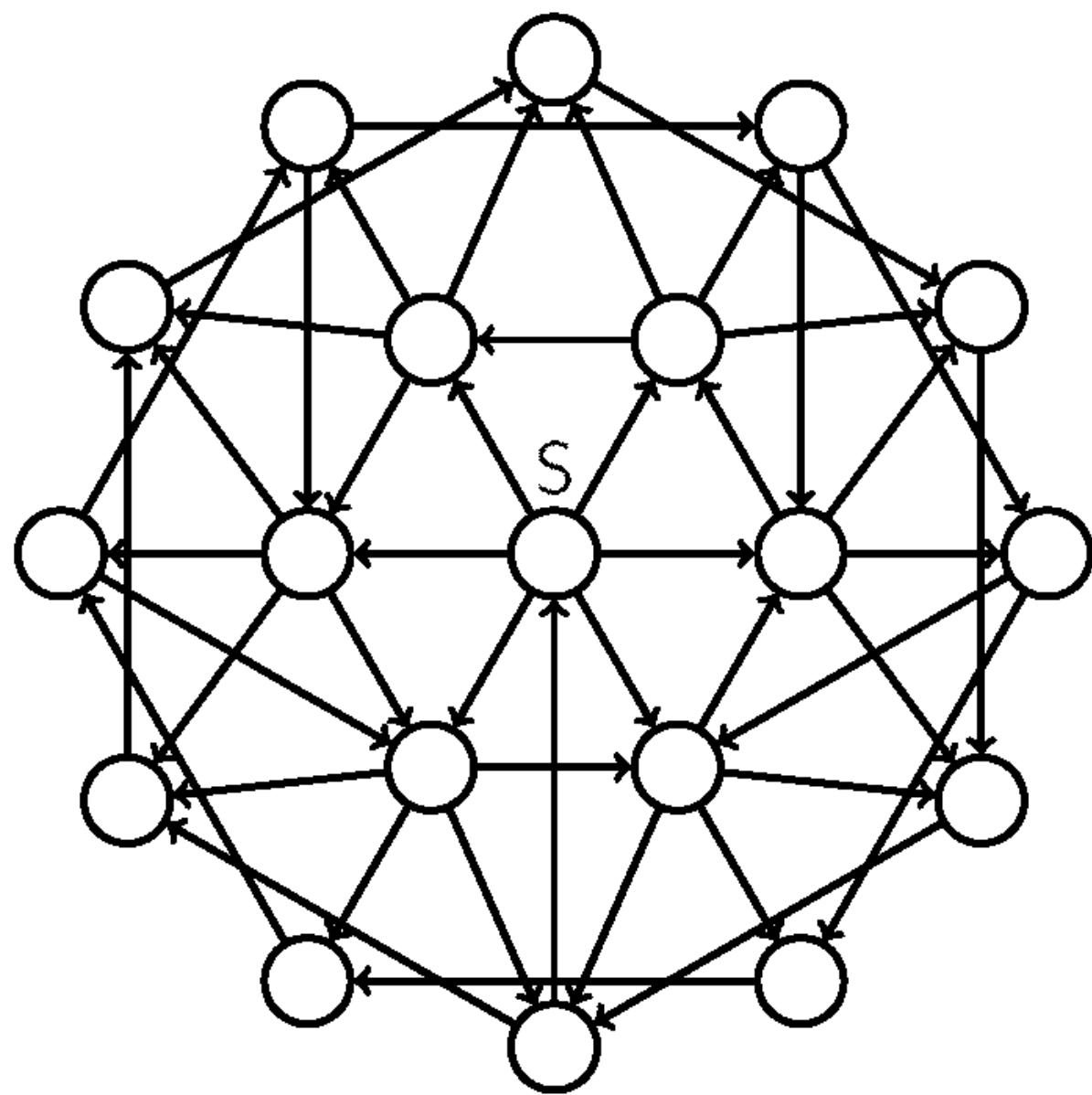
Starting from node S

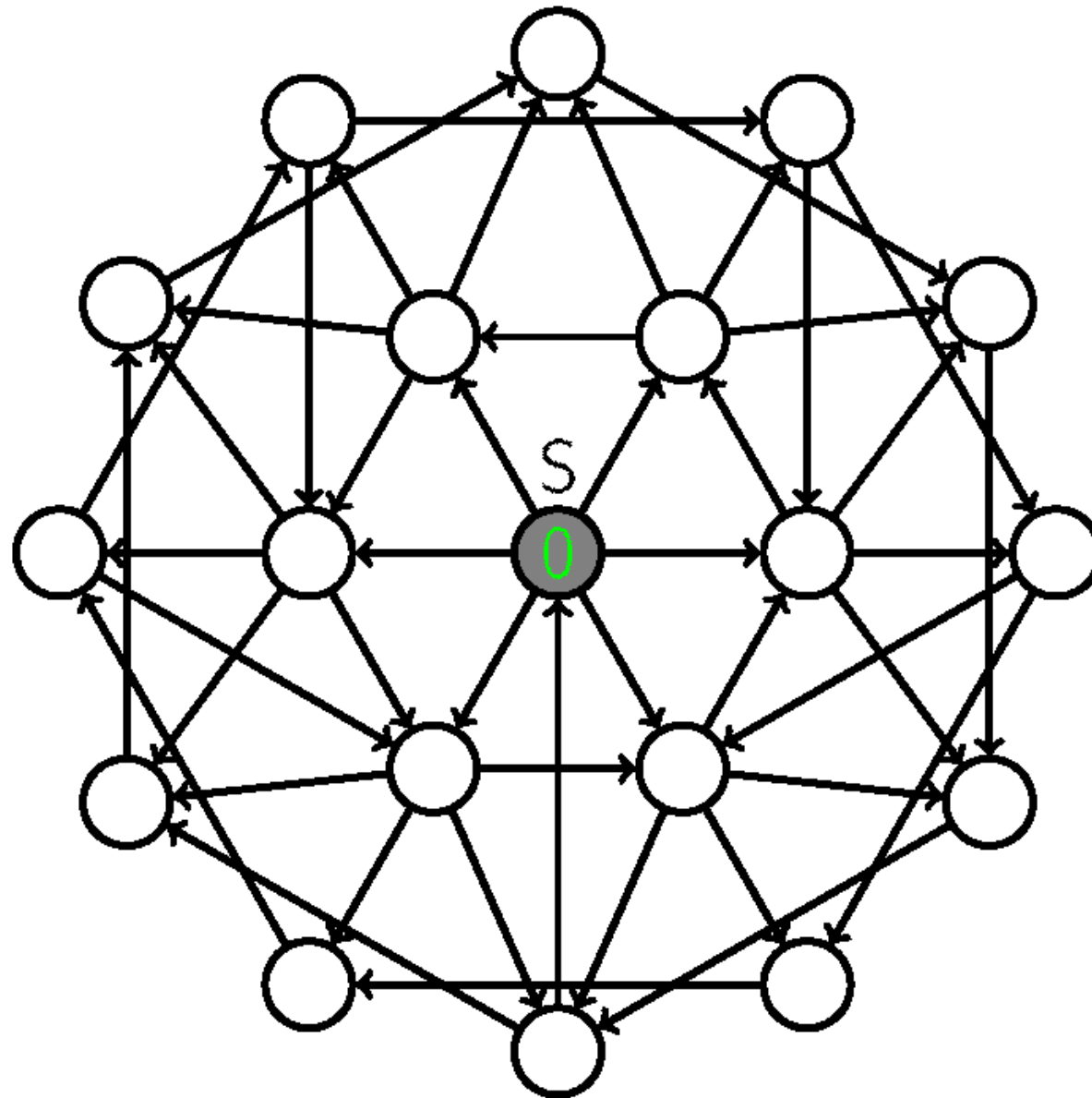
Breadth First Search for Calculating distance

BFS(G, S)

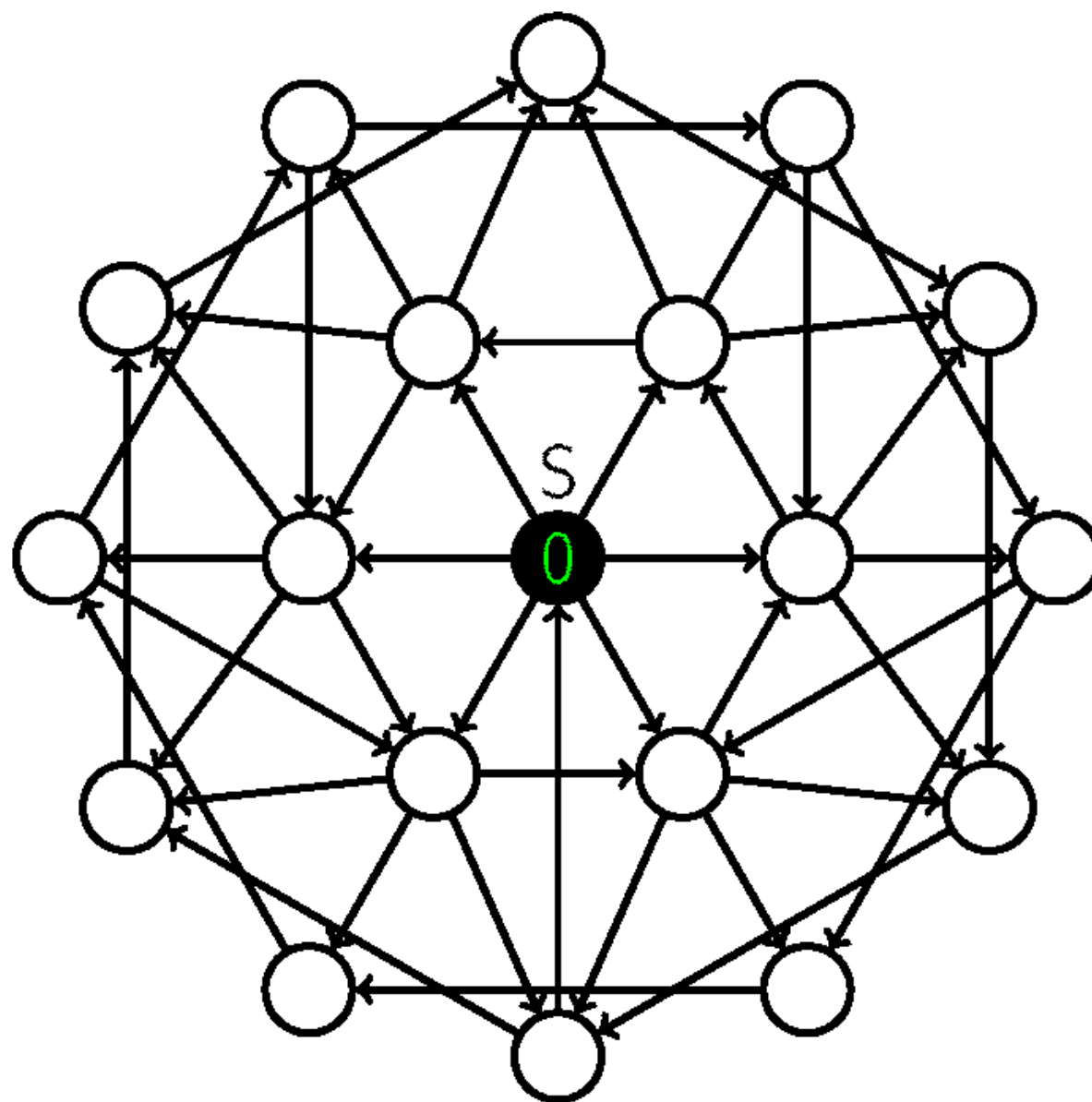
```
for all  $u \in V$ :  
     $\text{dist}[u] \leftarrow \infty$   
 $\text{dist}[S] \leftarrow 0$   
 $Q \leftarrow \{S\}$  {queue containing just  $S$ }  
while  $Q$  is not empty:  
     $u \leftarrow \text{Dequeue}(Q)$   
    for all  $(u, v) \in E$ :  
        if  $\text{dist}[v] = \infty$ :  
             $\text{Enqueue}(Q, v)$   
             $\text{dist}[v] \leftarrow \text{dist}[u] + 1$ 
```

Runtime
 $O(|E| + |V|)$

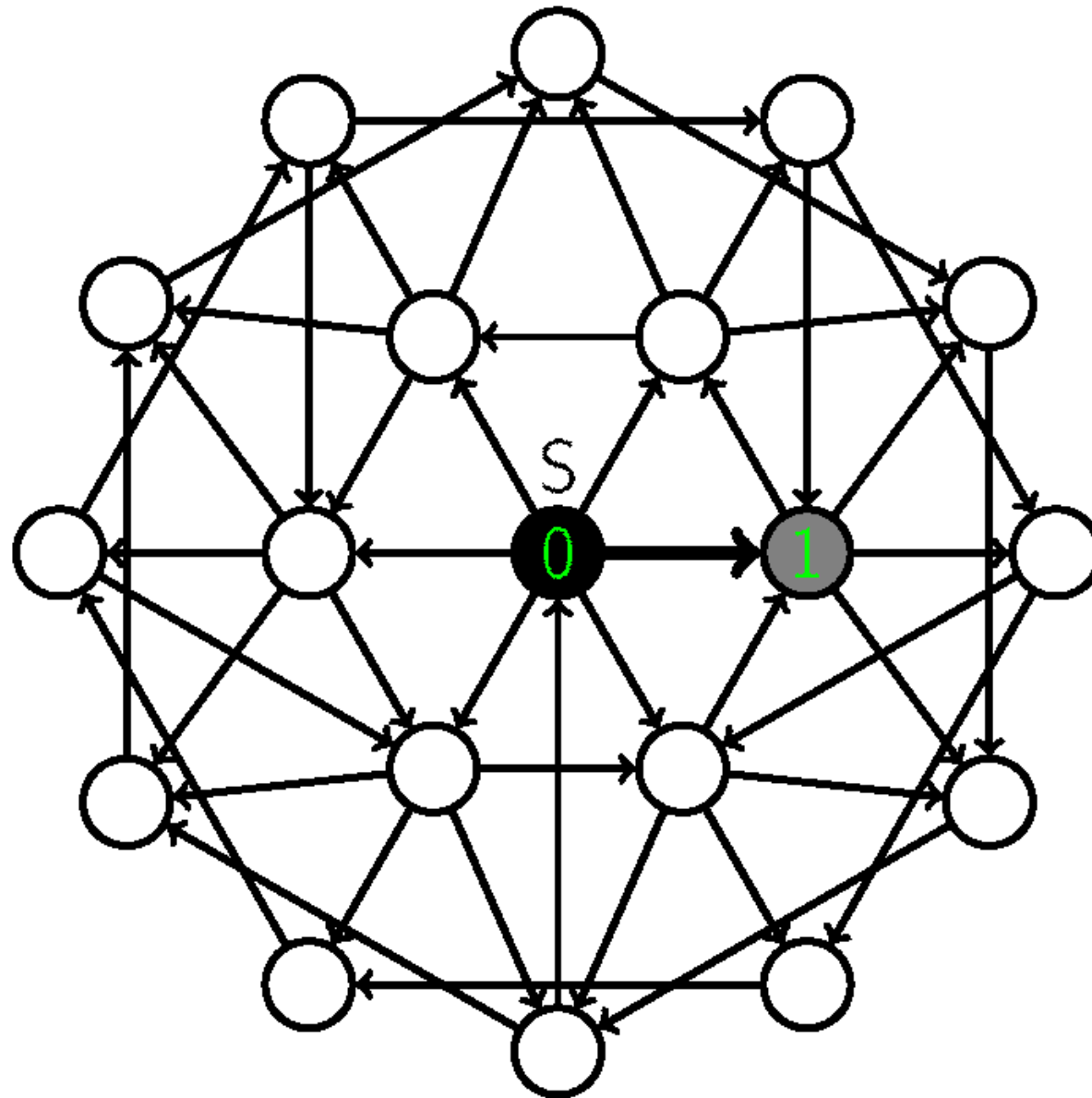




สืบทาแปลว่า Push node นั้นเข้าไปรอใน Queue (ถ้ายังค้างอยู่ใน Queue ก็จะทิ้งเป็นสืบทาไว้)
Node แรกสุดให้ dist = 0

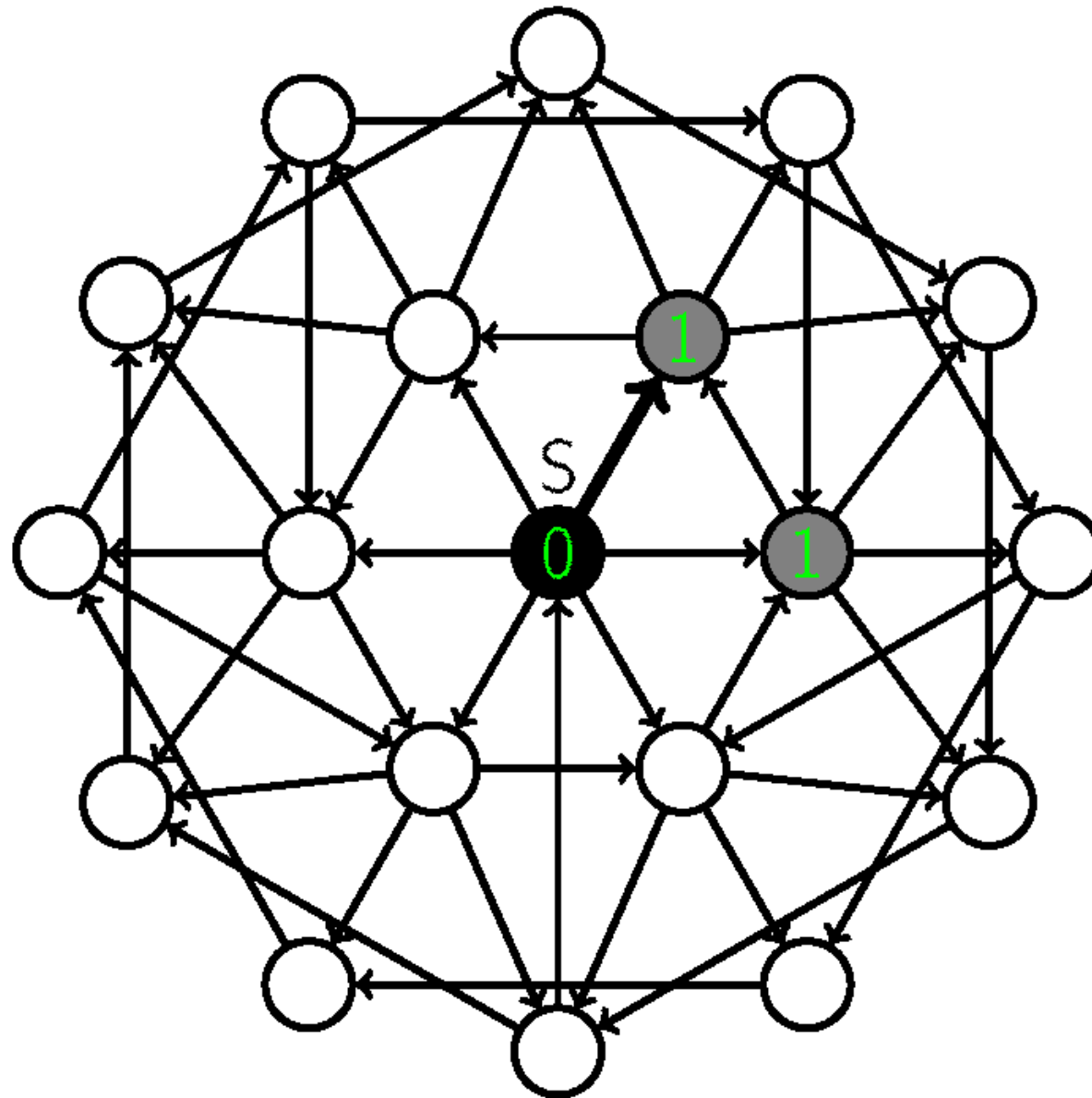


สืดำแปลว่า Pop node นั้น ออกจาก Queue เพื่อมาประมวลผล (ถ้าประมวลผลแล้วก็จะทิ้งเป็นสืดำไว้)

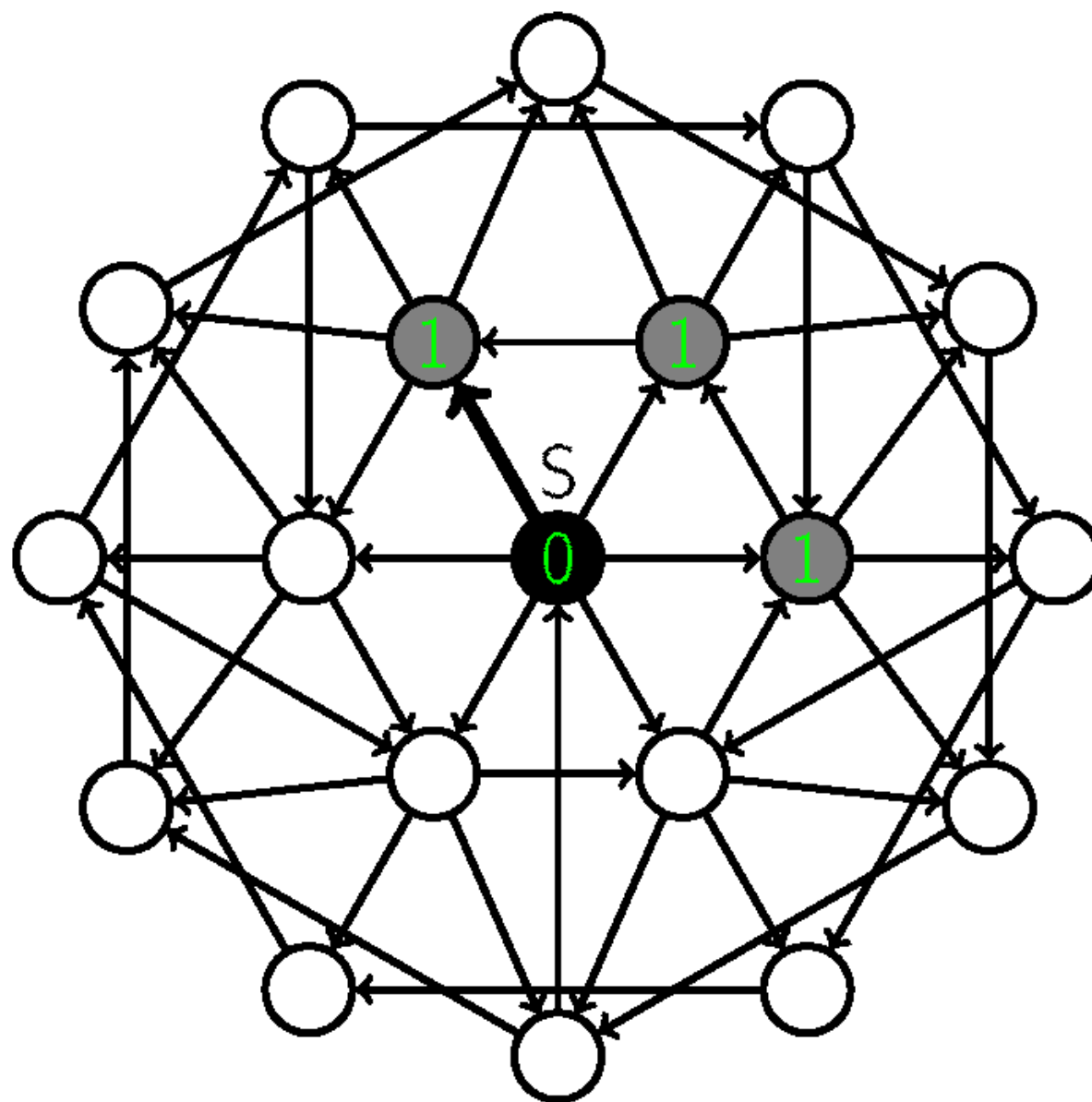


พิจารณา Edge ทุกเส้นที่ต่ออยู่กับ S แล้วให้ดูปลายทางว่า $dist$ มีค่าแล้วหรือยัง

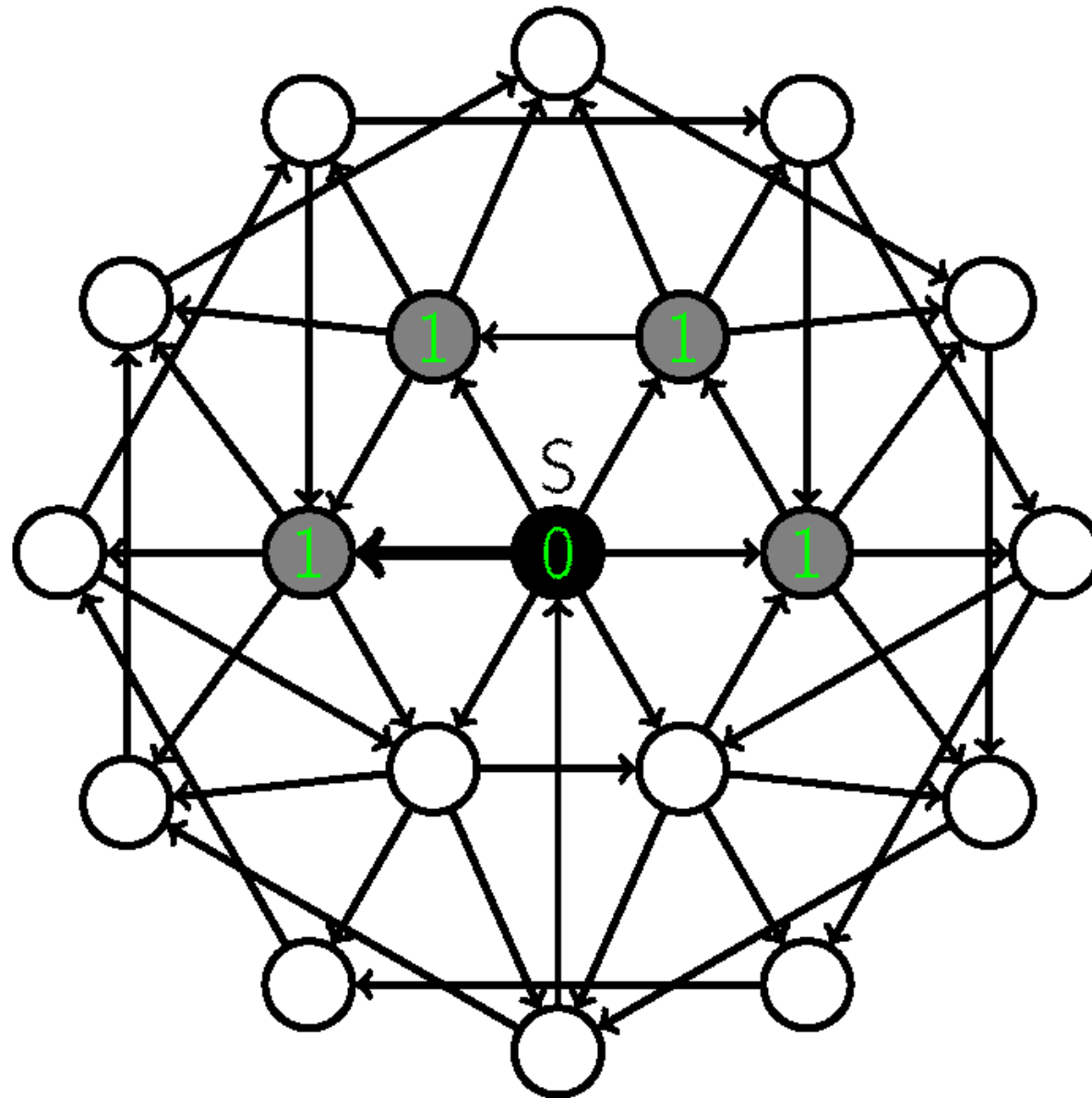
ถ้ายัง ก็ให้ Push Node ปลายทางนั้นเข้าไปใน Queue พร้อมกับการเซต $dist(v)$ เท่ากับ $dist(u)+1$ เสมอ



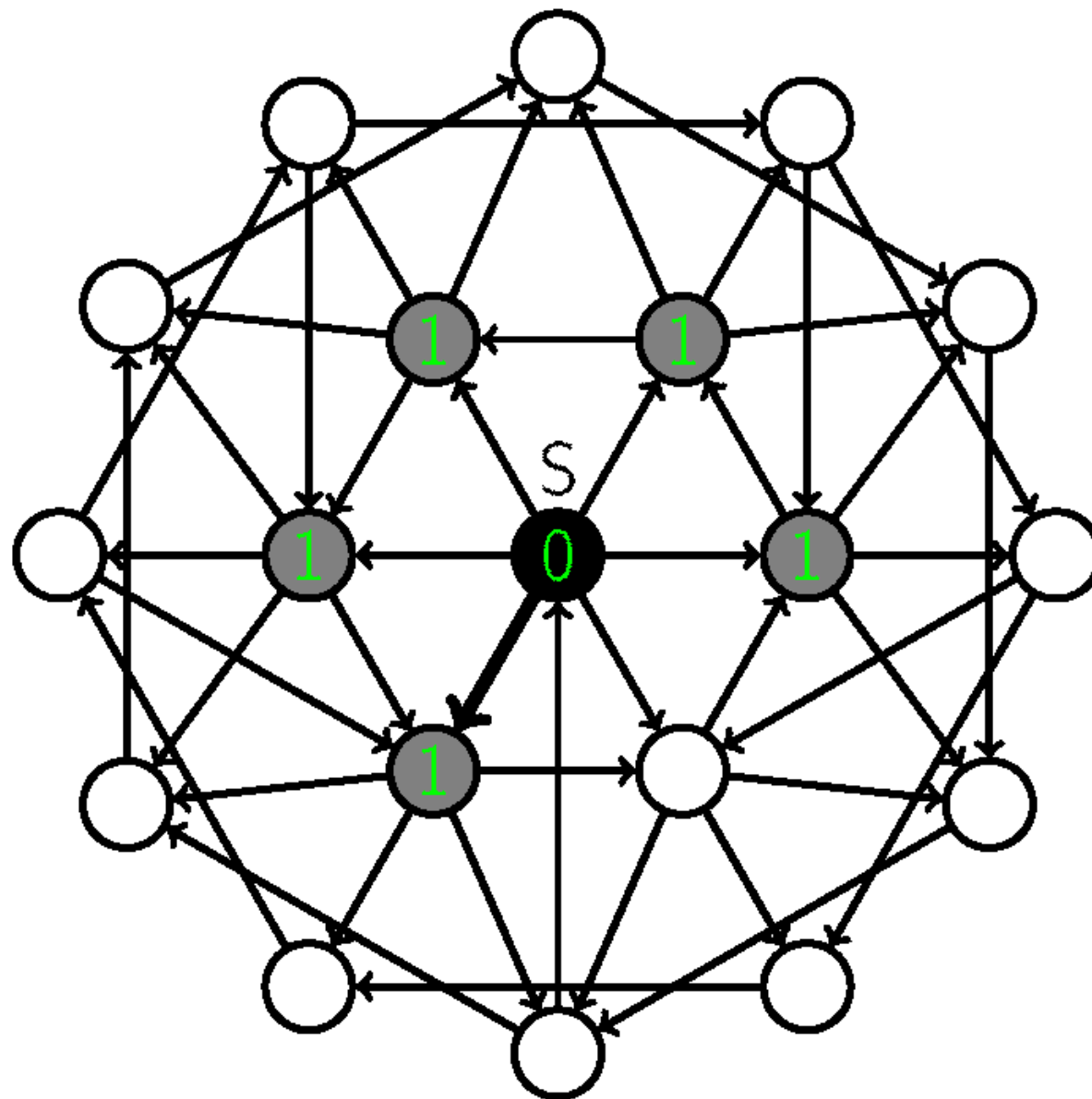
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$



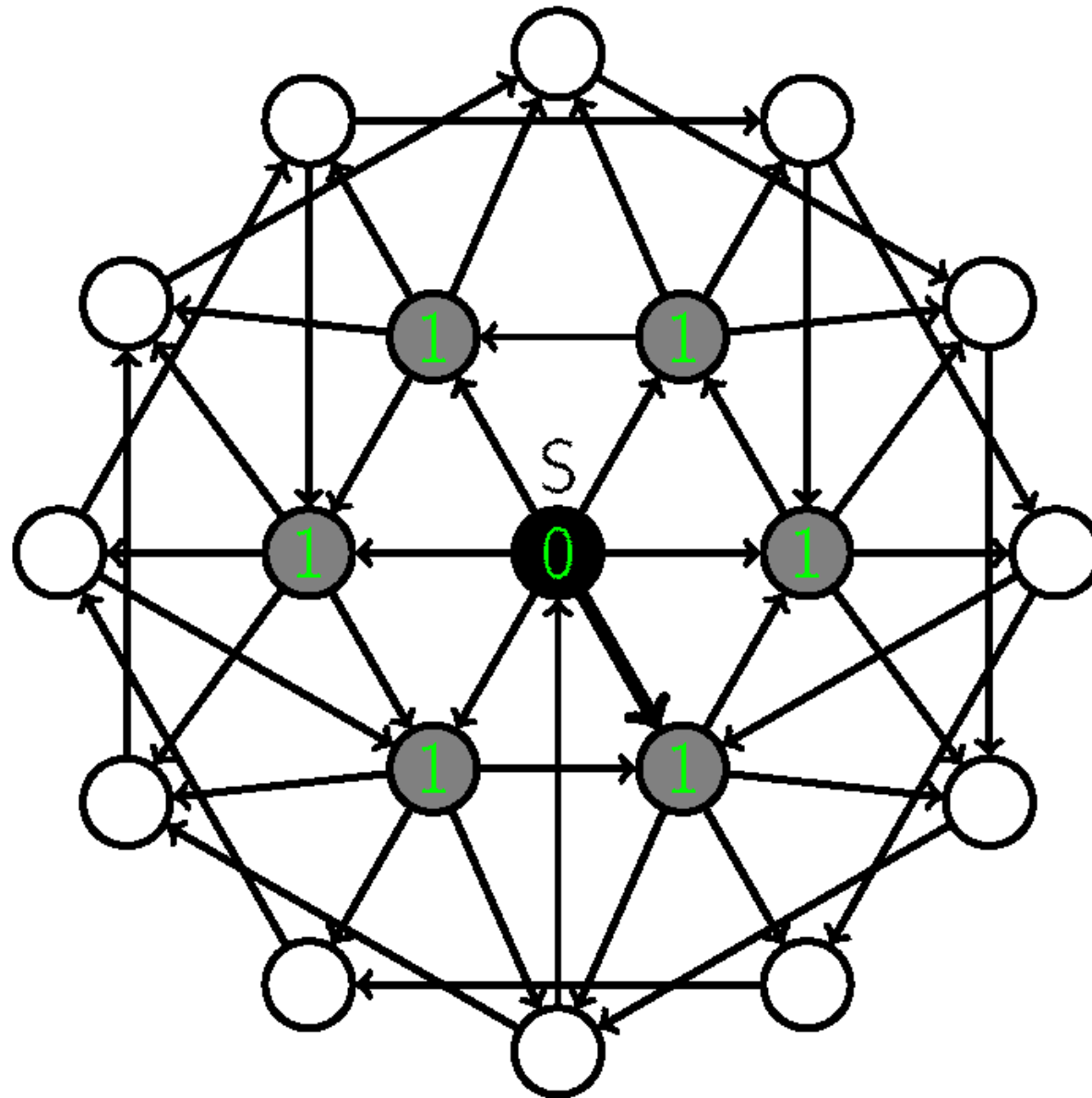
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$



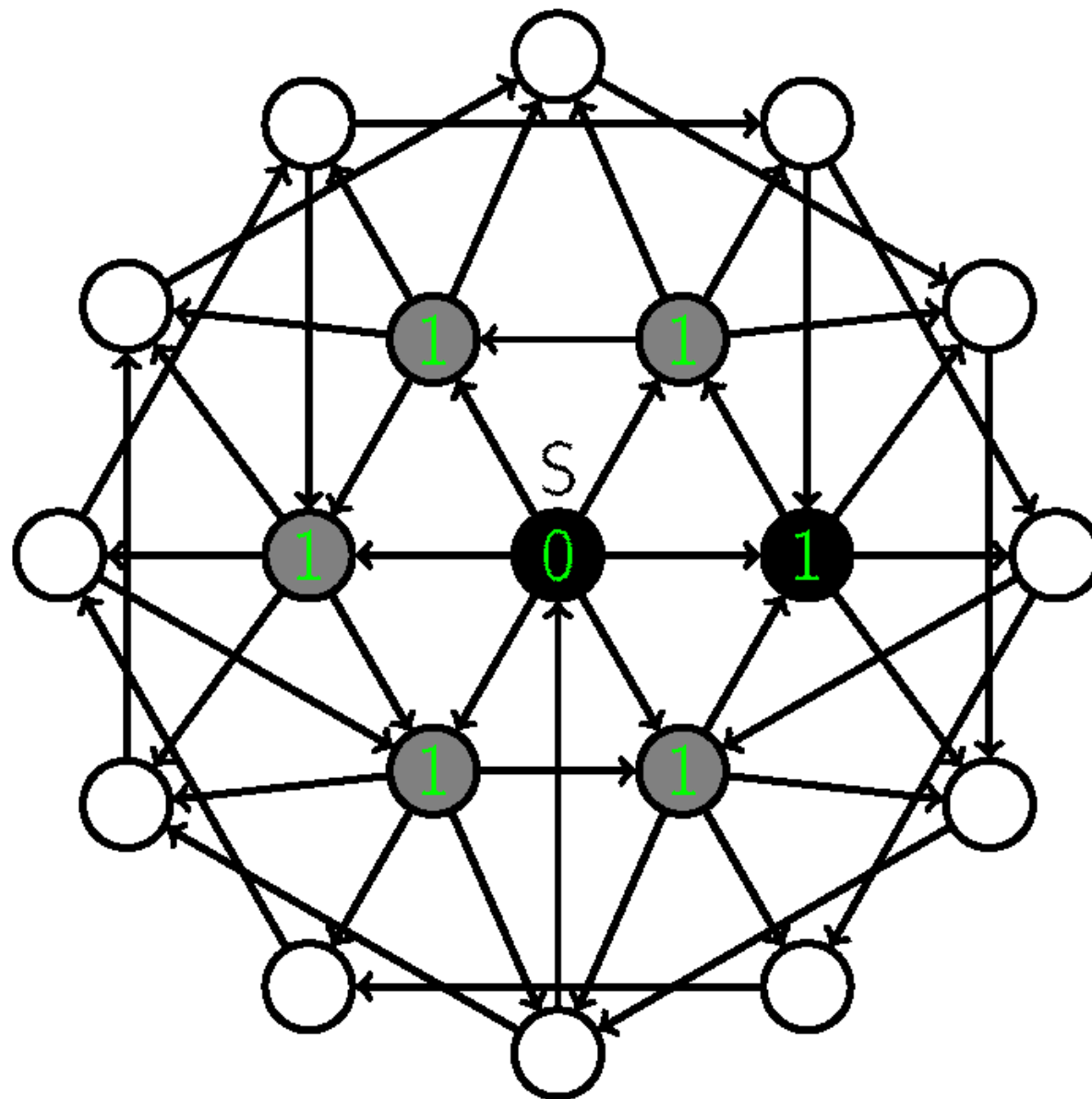
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า dist แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $\text{dist}(v) = \text{dist}(u) + 1$



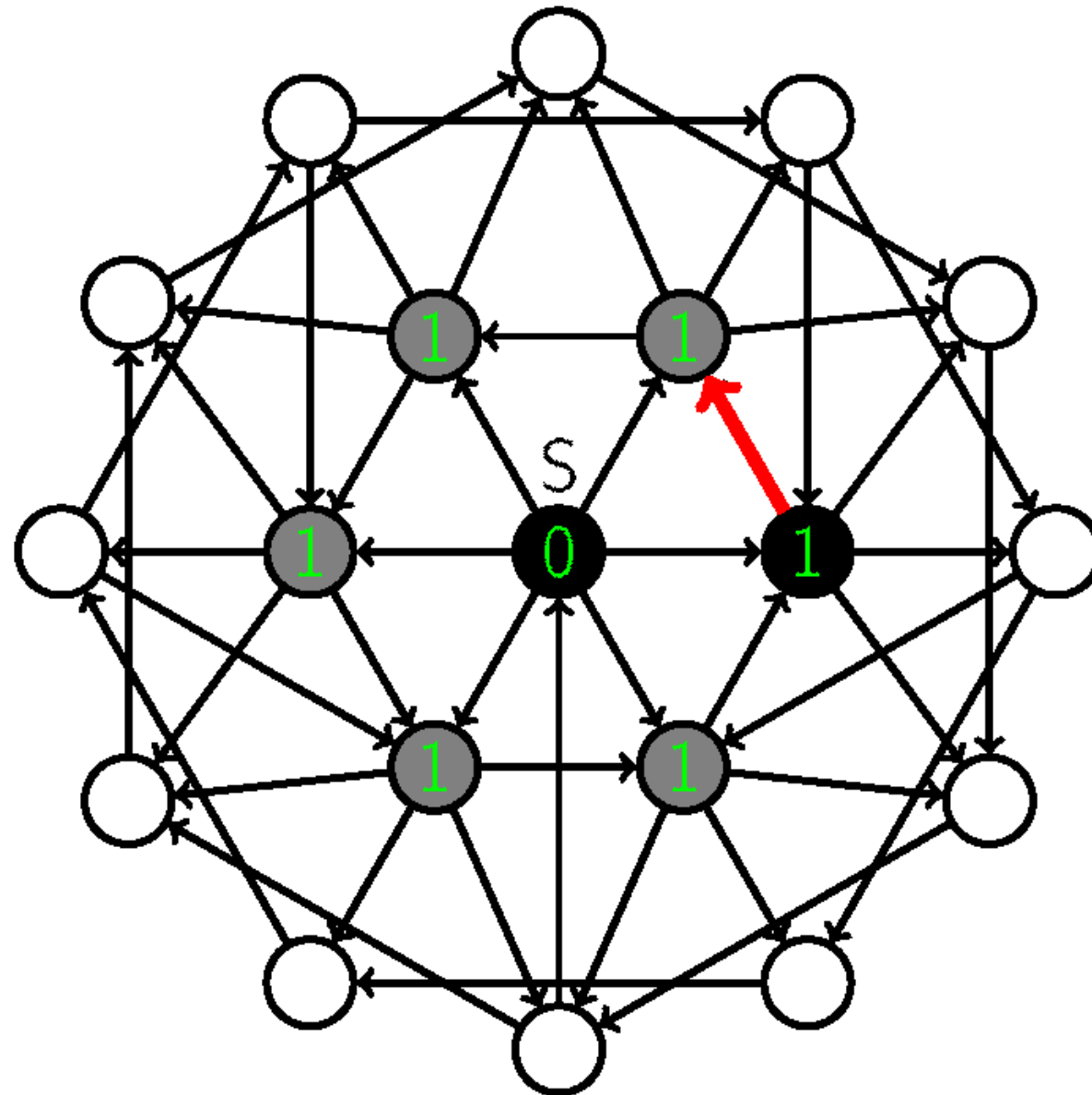
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$



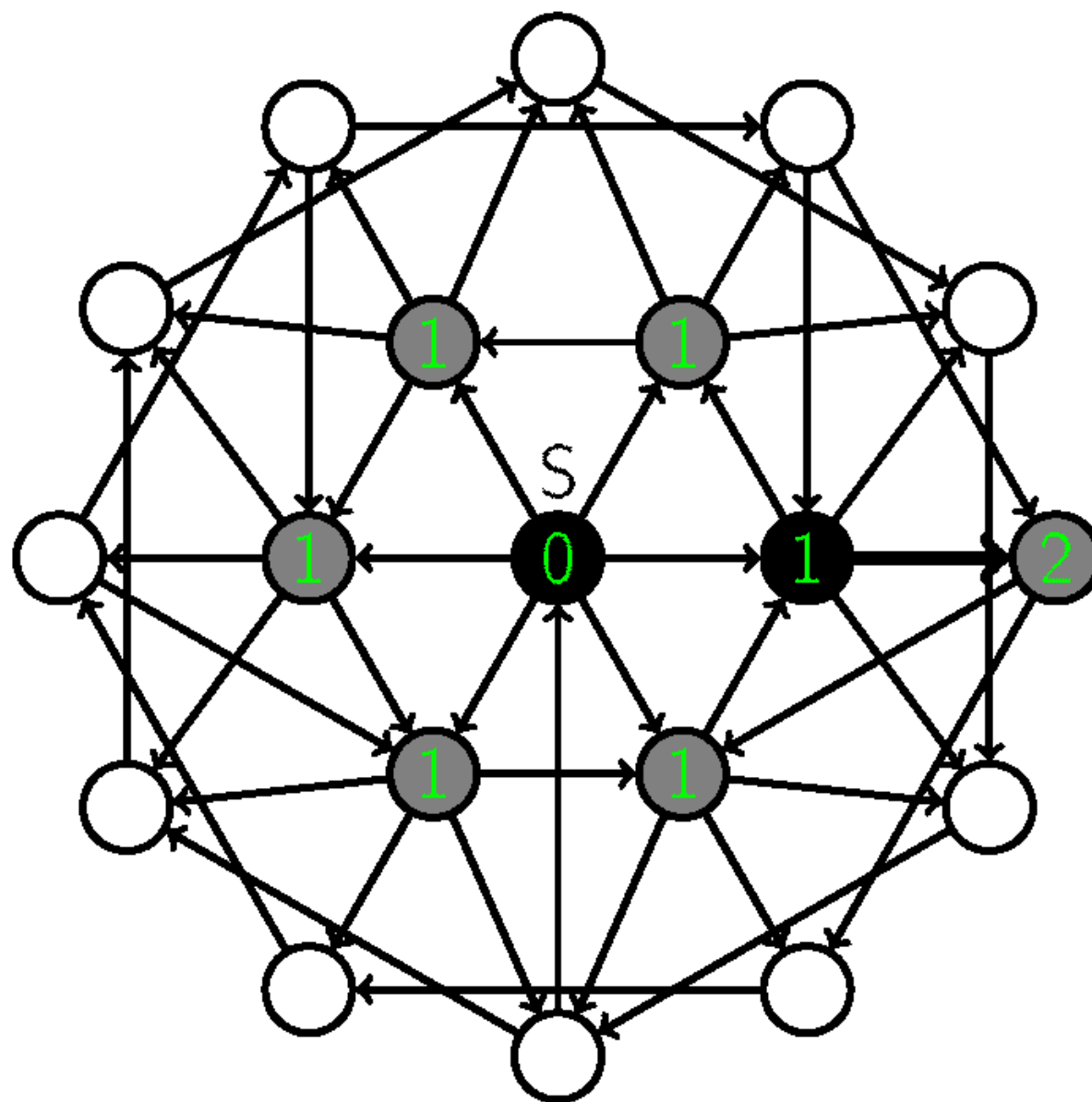
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$



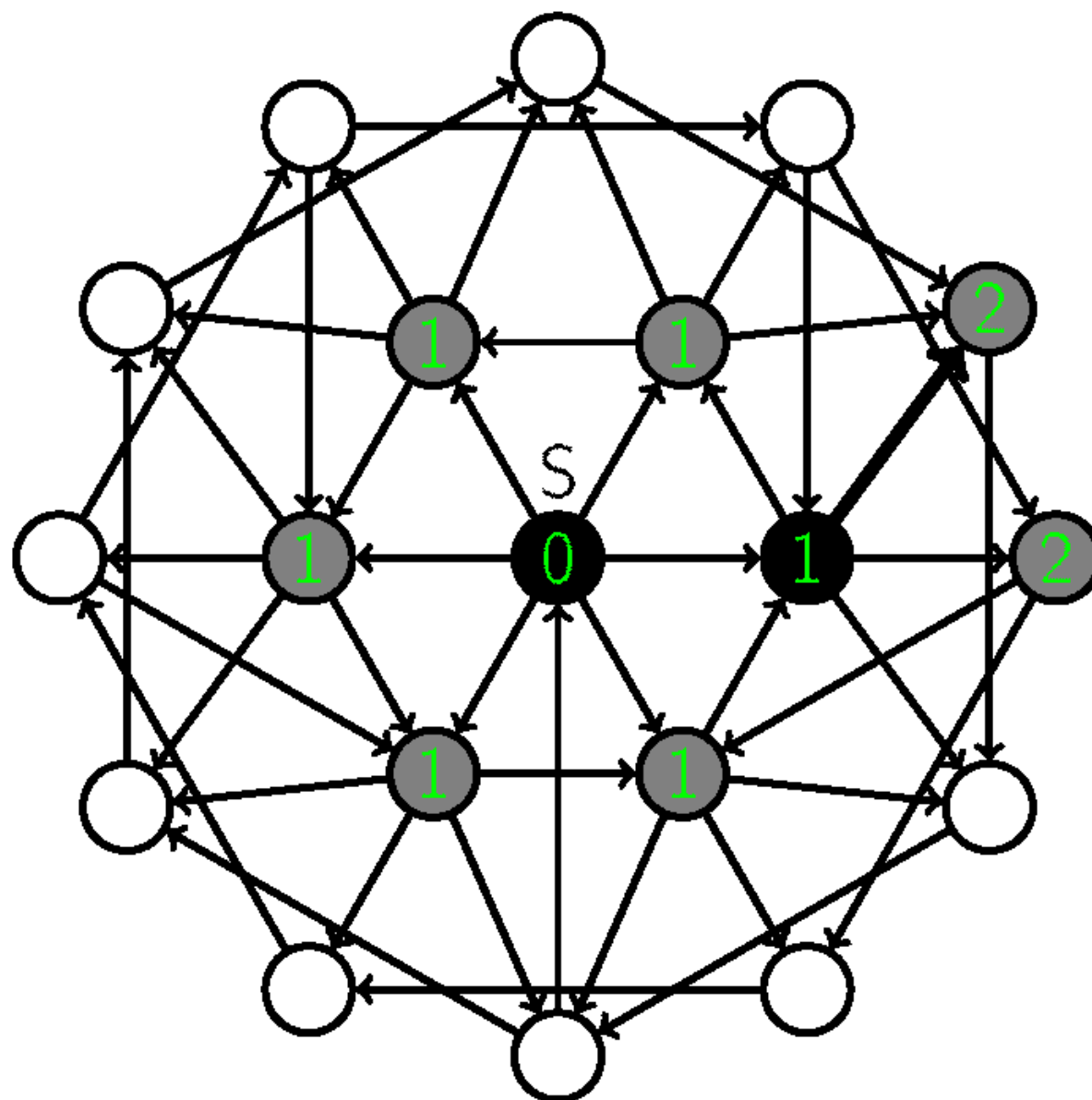
เมื่อ Edge ทุกเส้นของ S ถูกประมวลผลครบแล้ว ก็สิ้นสุดกระบวนการของ S
 Pop Queue ออกมาแล้วเริ่มกระบวนการใหม่ นั่นก็คือ ตรวจสอบ Edge ทุกเส้นที่เชื่อมต่อ ว่าปลายทางมี dist แล้วหรือยัง



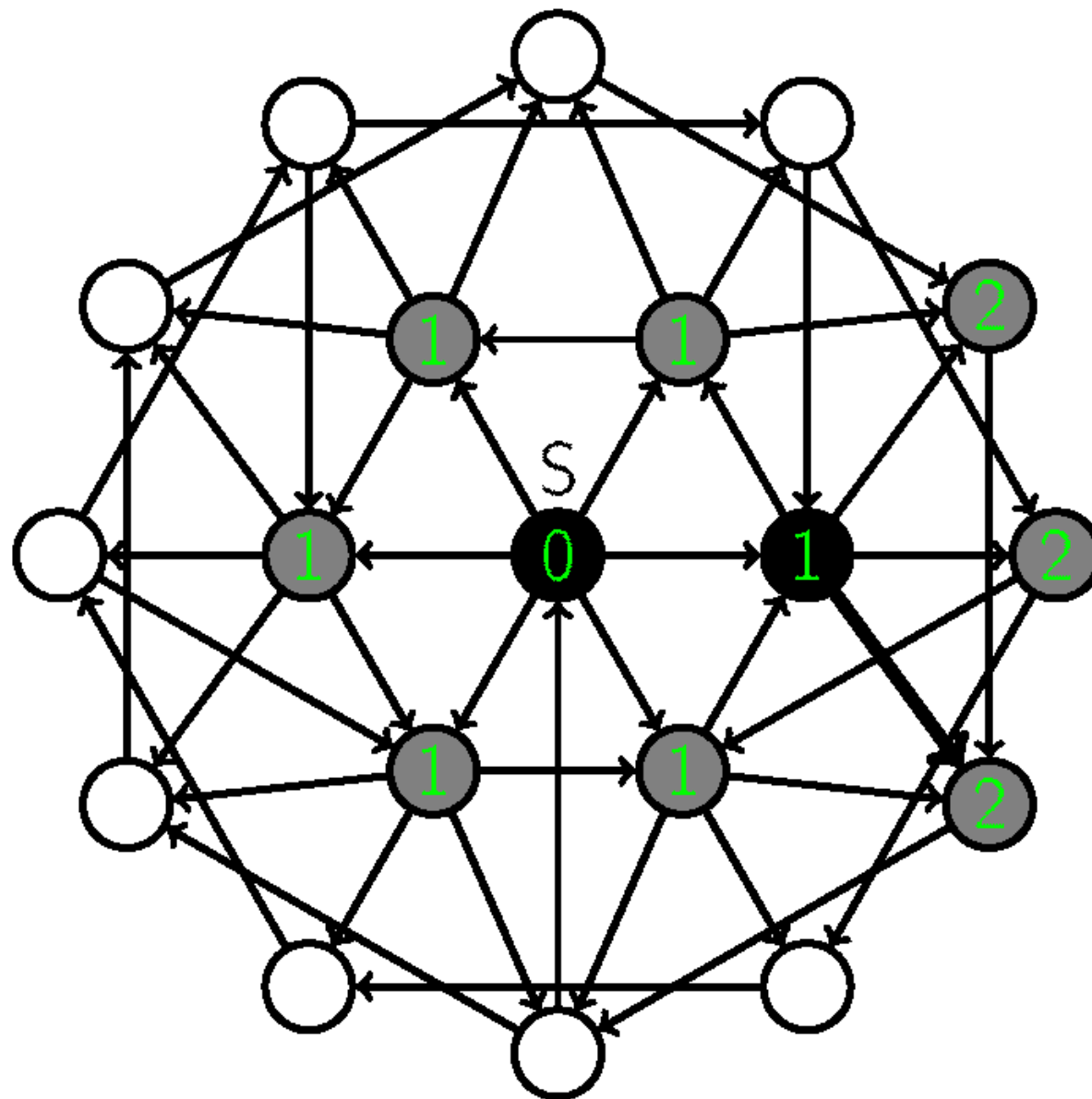
เส้นสีแดงแปลว่า ปลายทางมีค่า dist แล้ว เพราะฉะนั้นข้าม ไปดู Edge เส้นอื่น



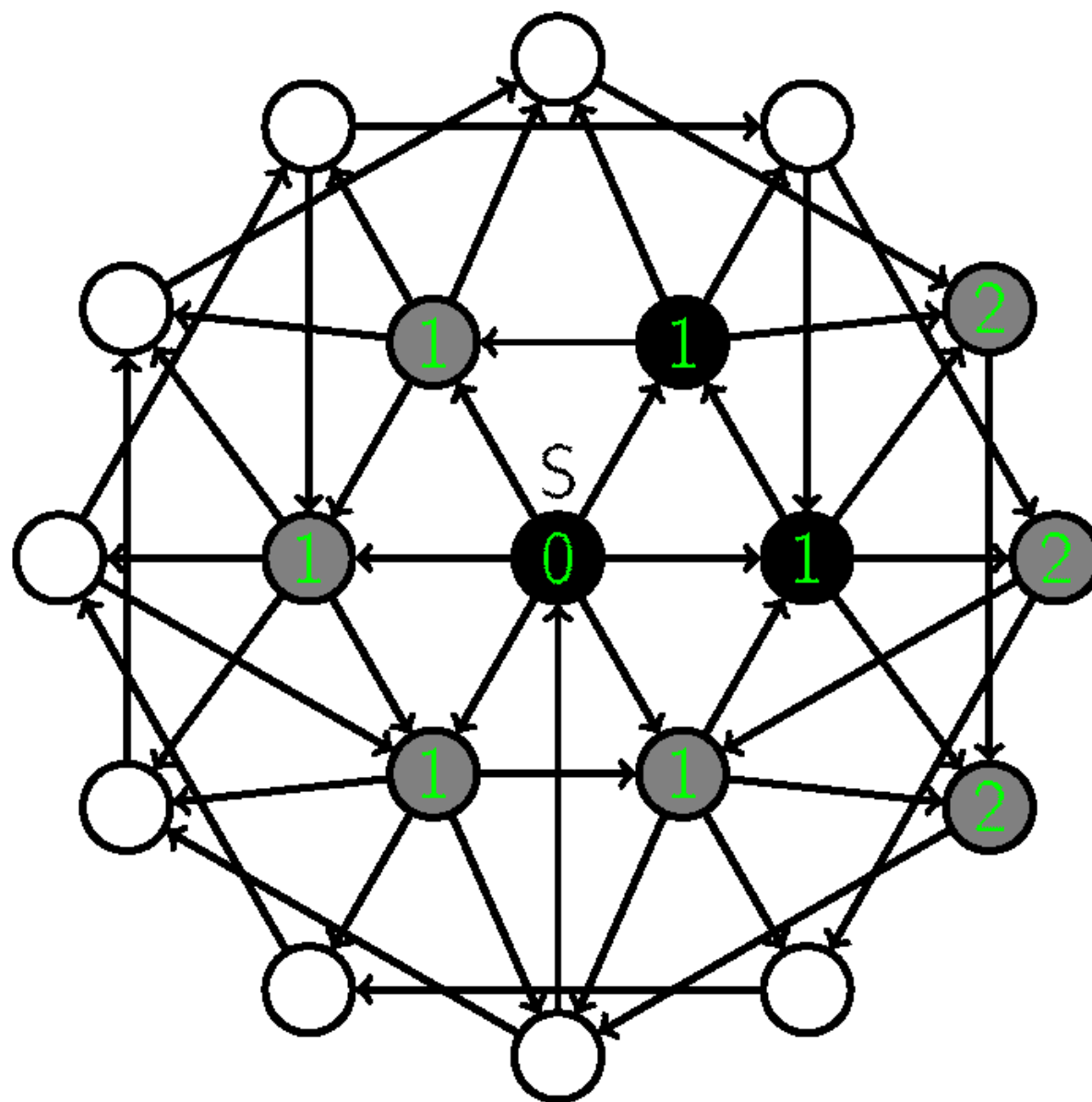
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$



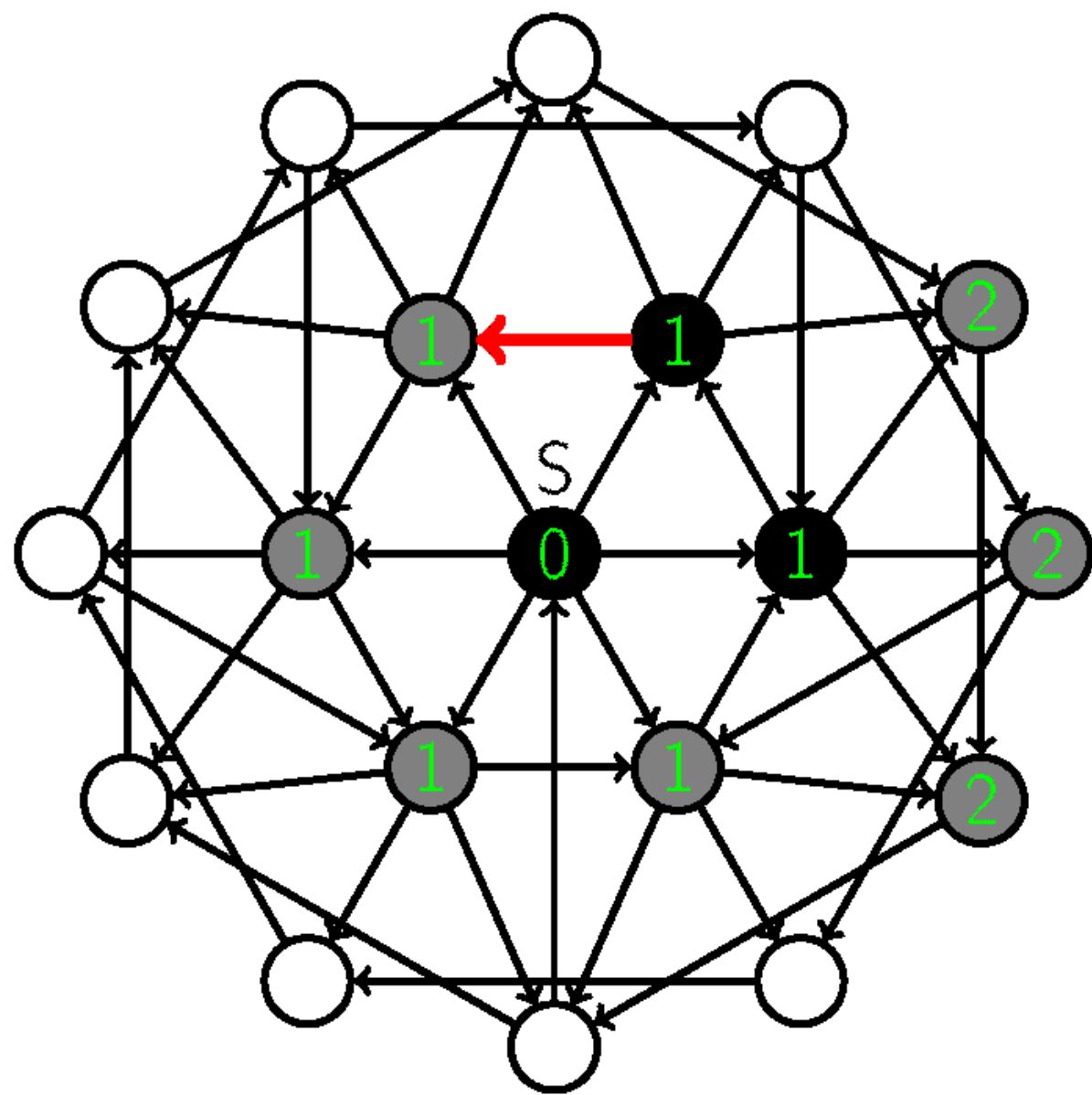
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า $dist$ แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $dist(v) = dist(u) + 1$

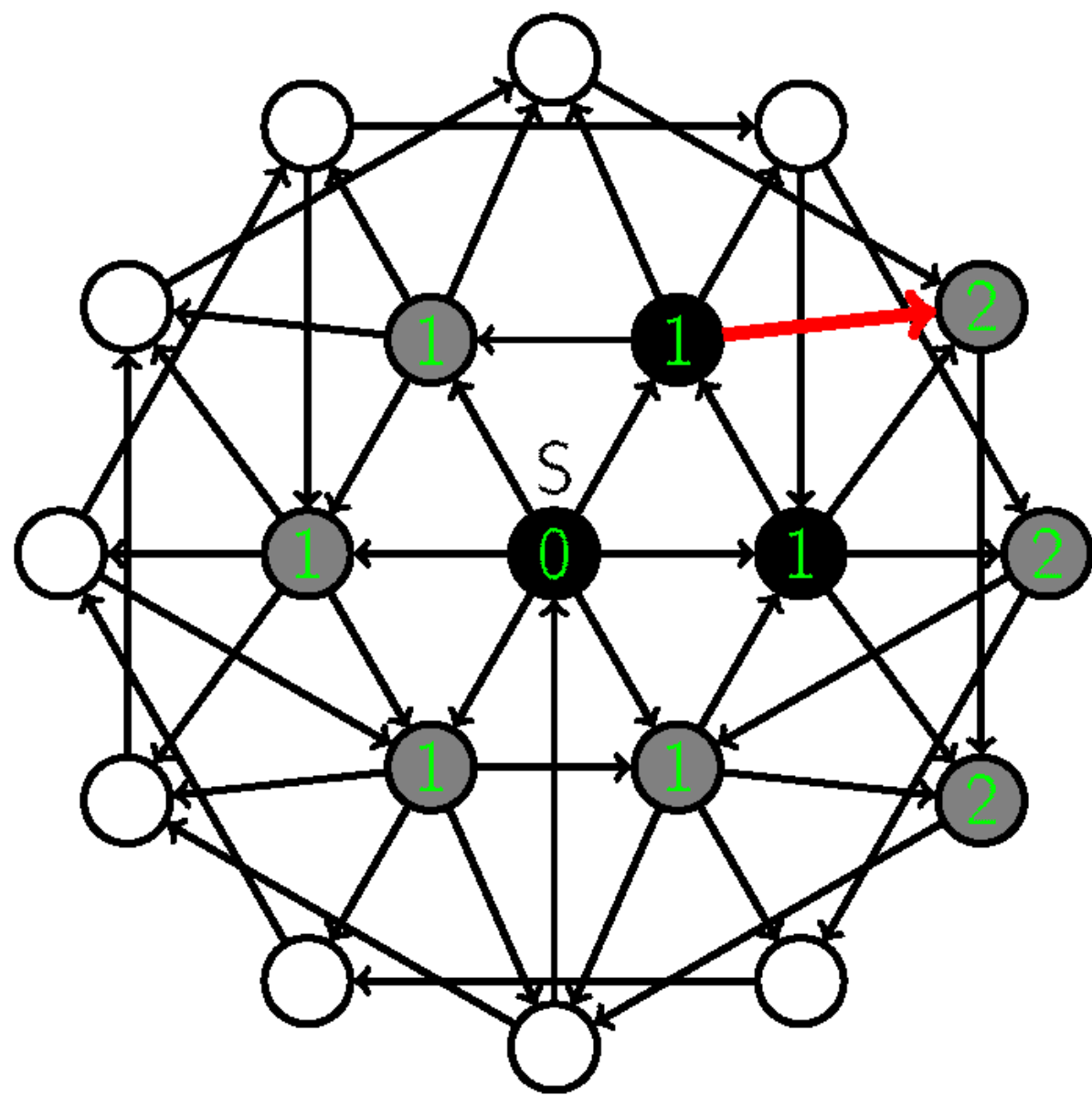


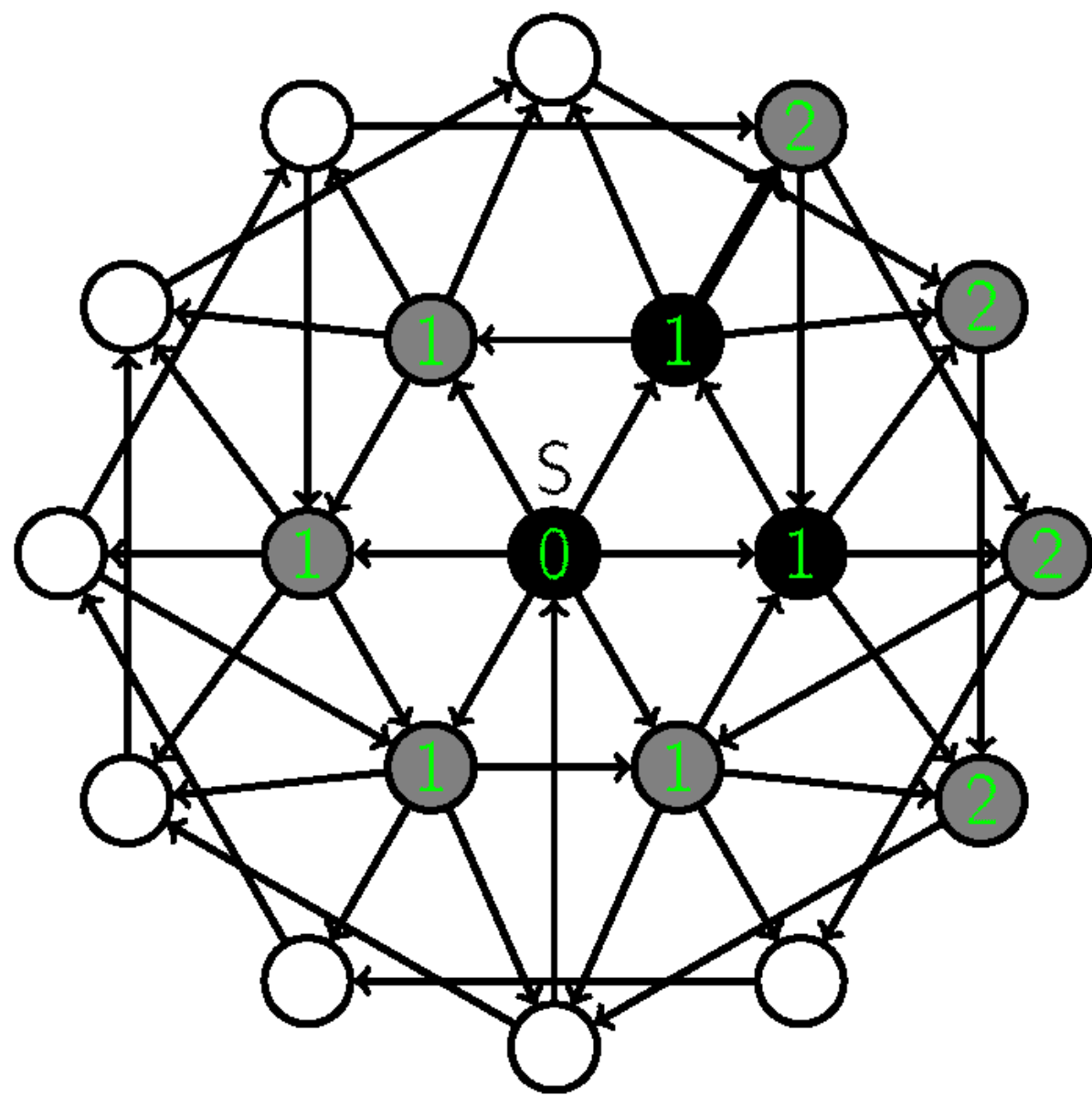
พิจารณาในทุก ๆ Edge ของ S ว่าต่อกับอะไรอีก แล้วเช็คค่า Node ปลายทางมีค่า dist แล้วหรือยัง
ถ้ายังก็ Push เข้าไปใน Queue แล้วเซต $\text{dist}(v) = \text{dist}(u) + 1$

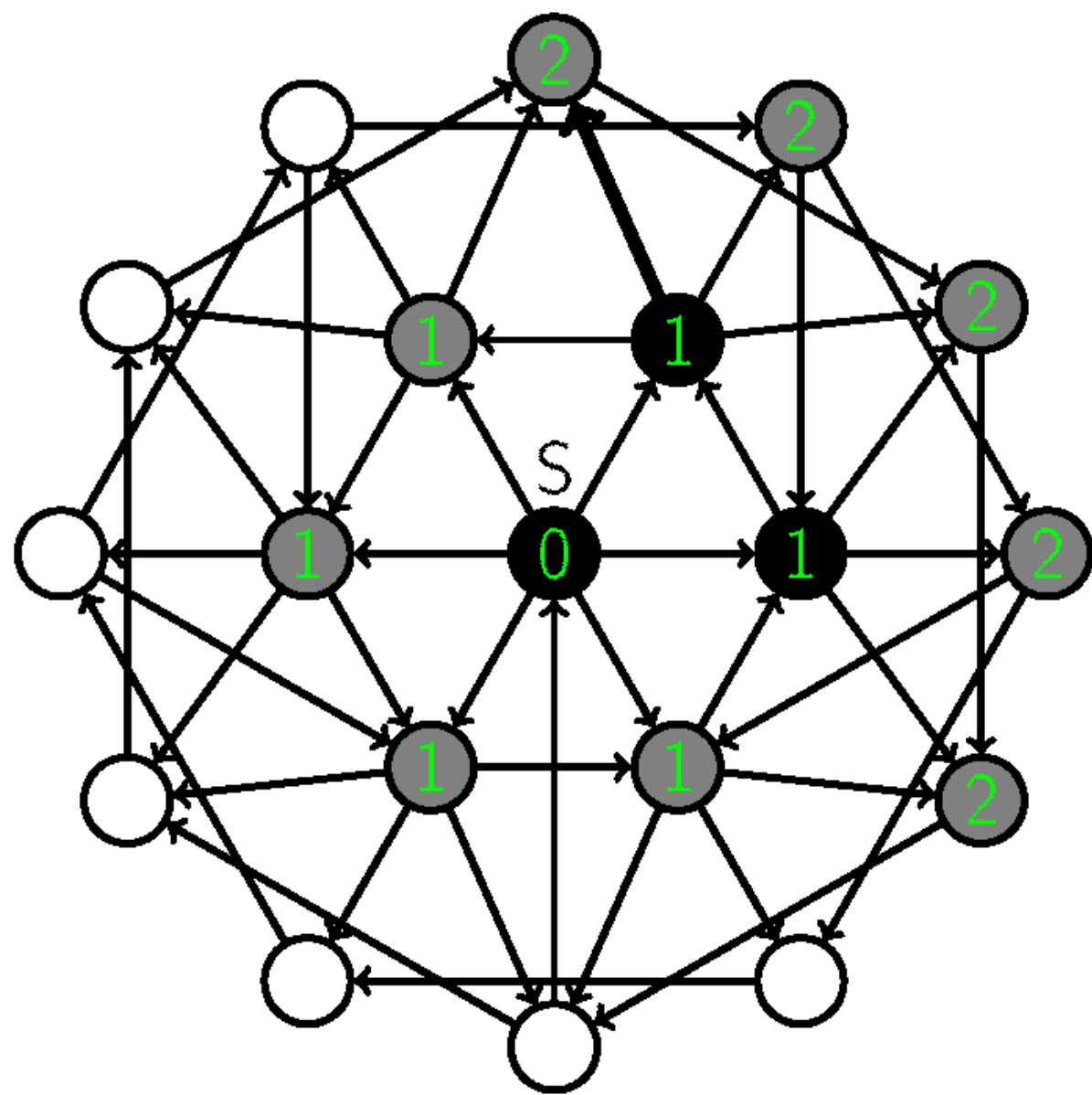


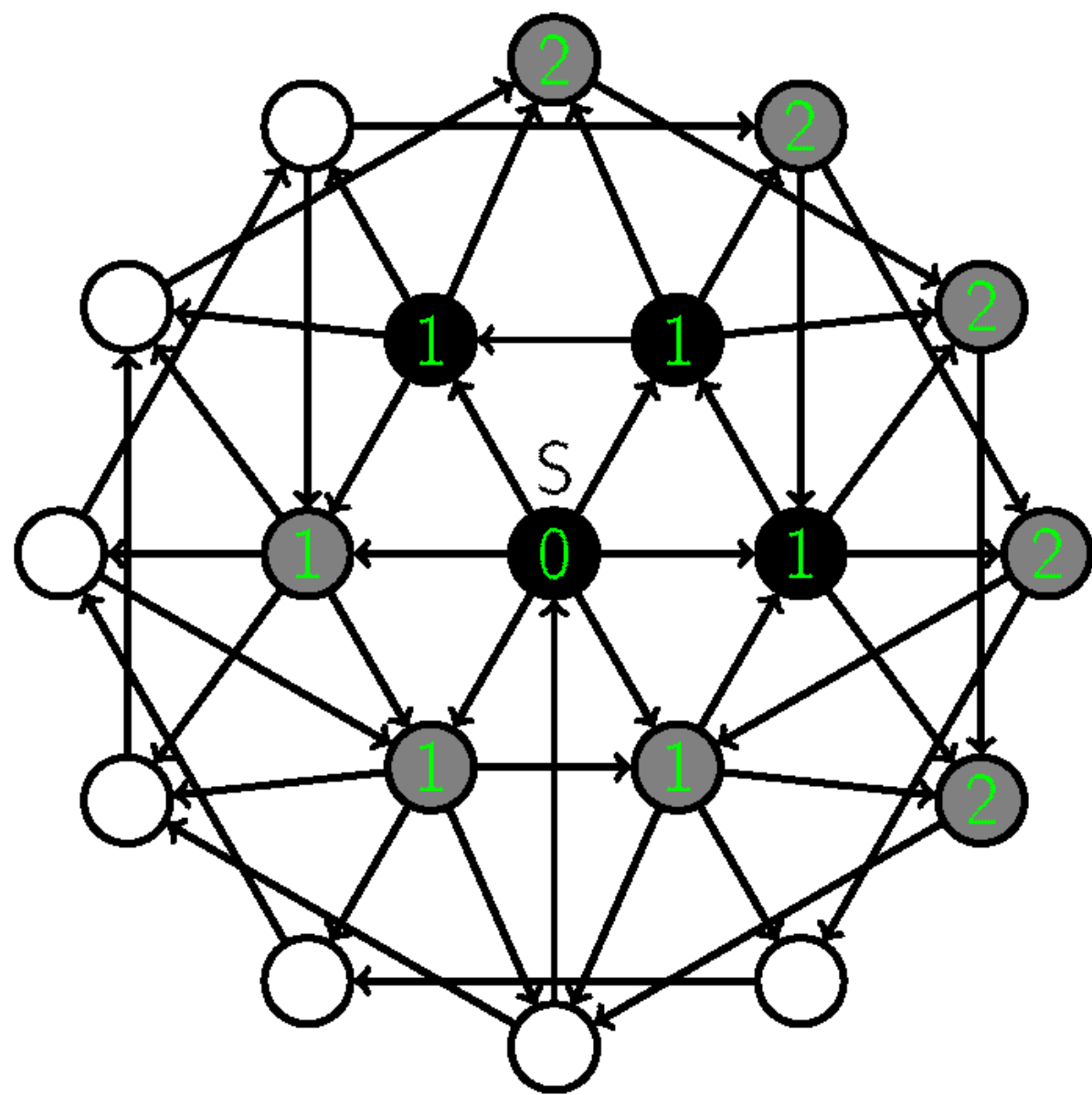
เมื่อ Edge ทุกเส้นของ Node ถูกประมวลผลครบแล้ว ก็สิ้นสุดกระบวนการ
 Pop Queue ออกมาแล้วเริ่มกระบวนการใหม่ นั่นก็คือ ตรวจสอบ Edge ทุกเส้นที่เชื่อมต่อ ว่าปลายทางมี dist แล้วหรือยัง

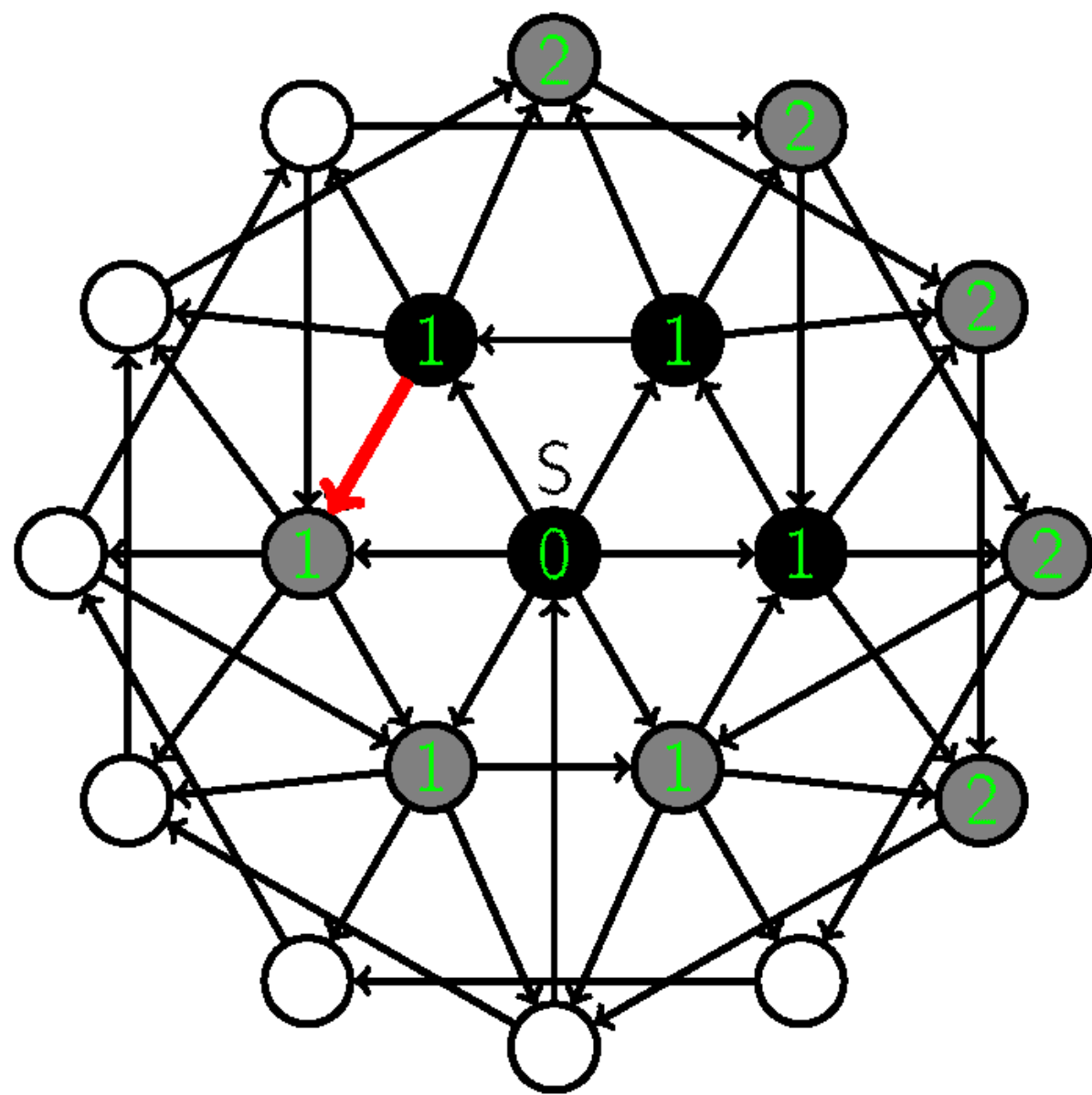


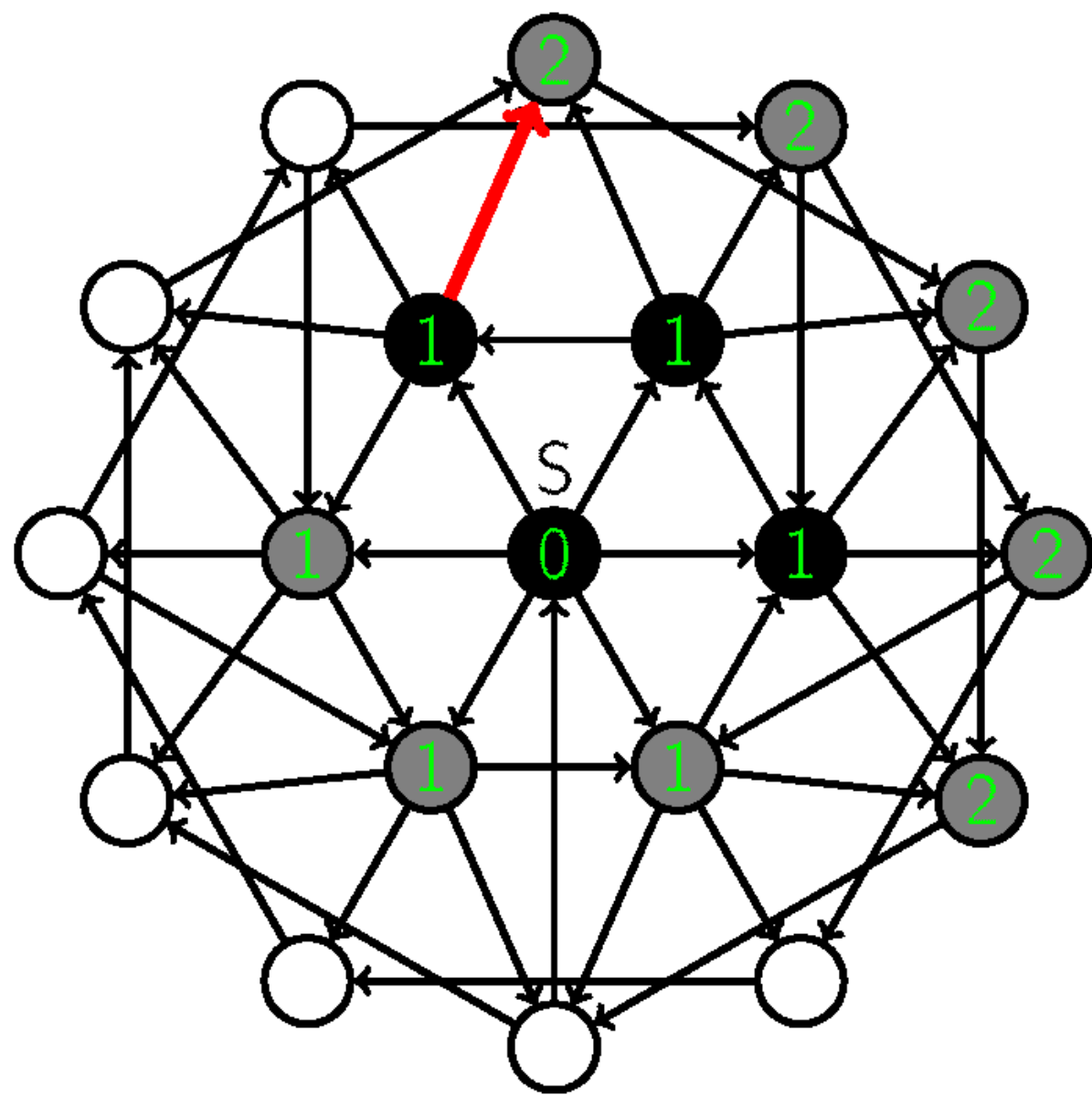


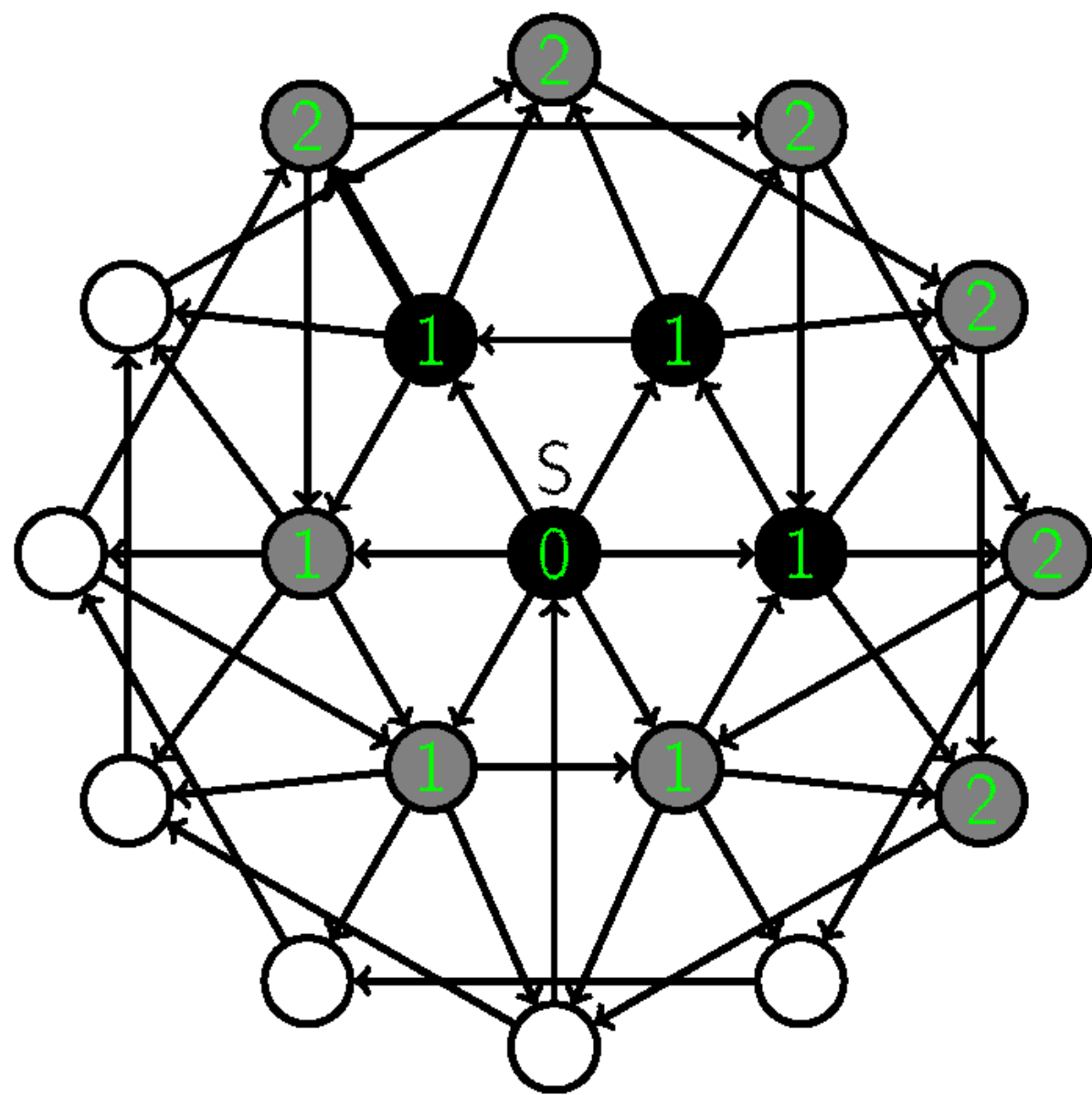


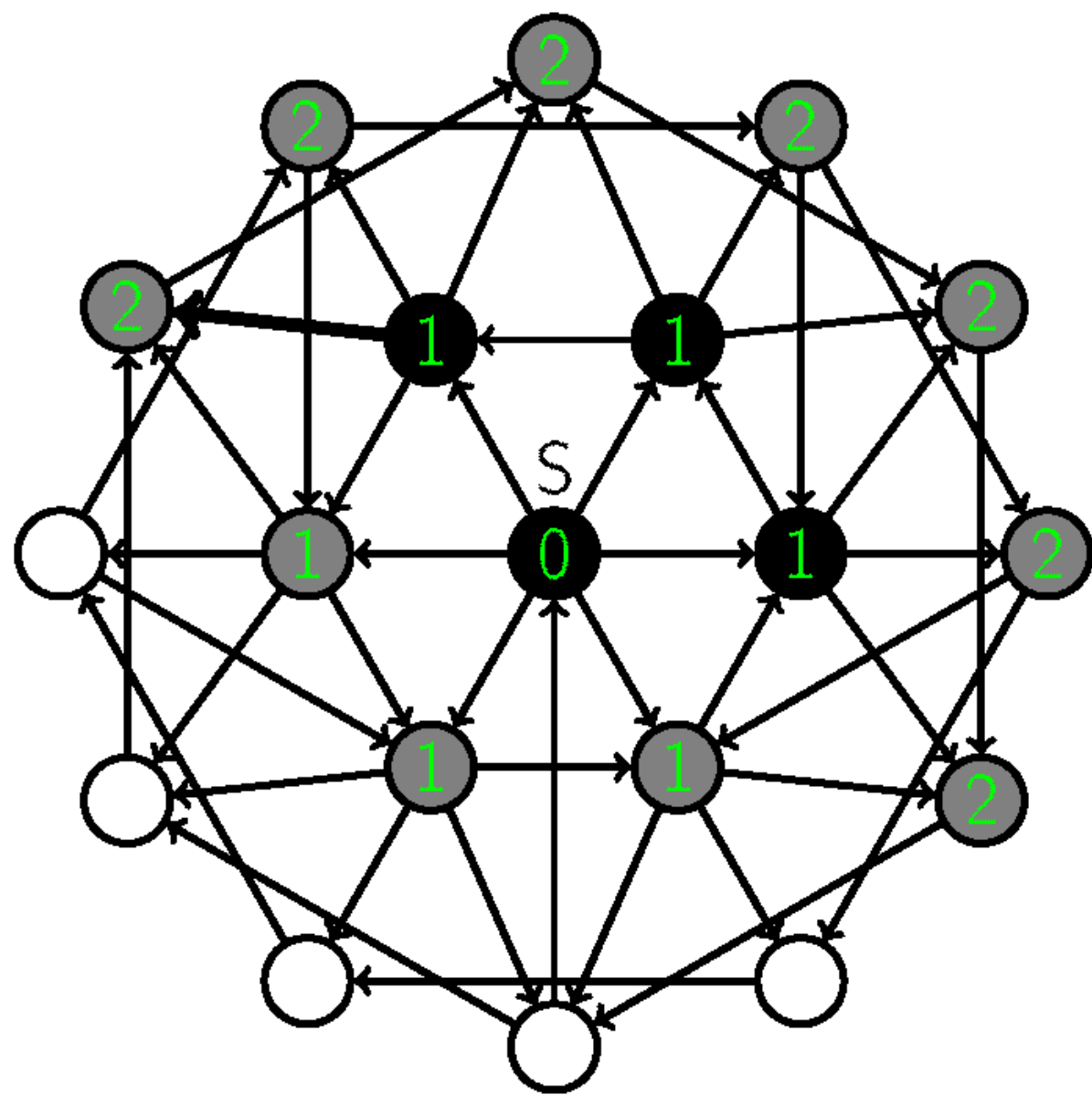


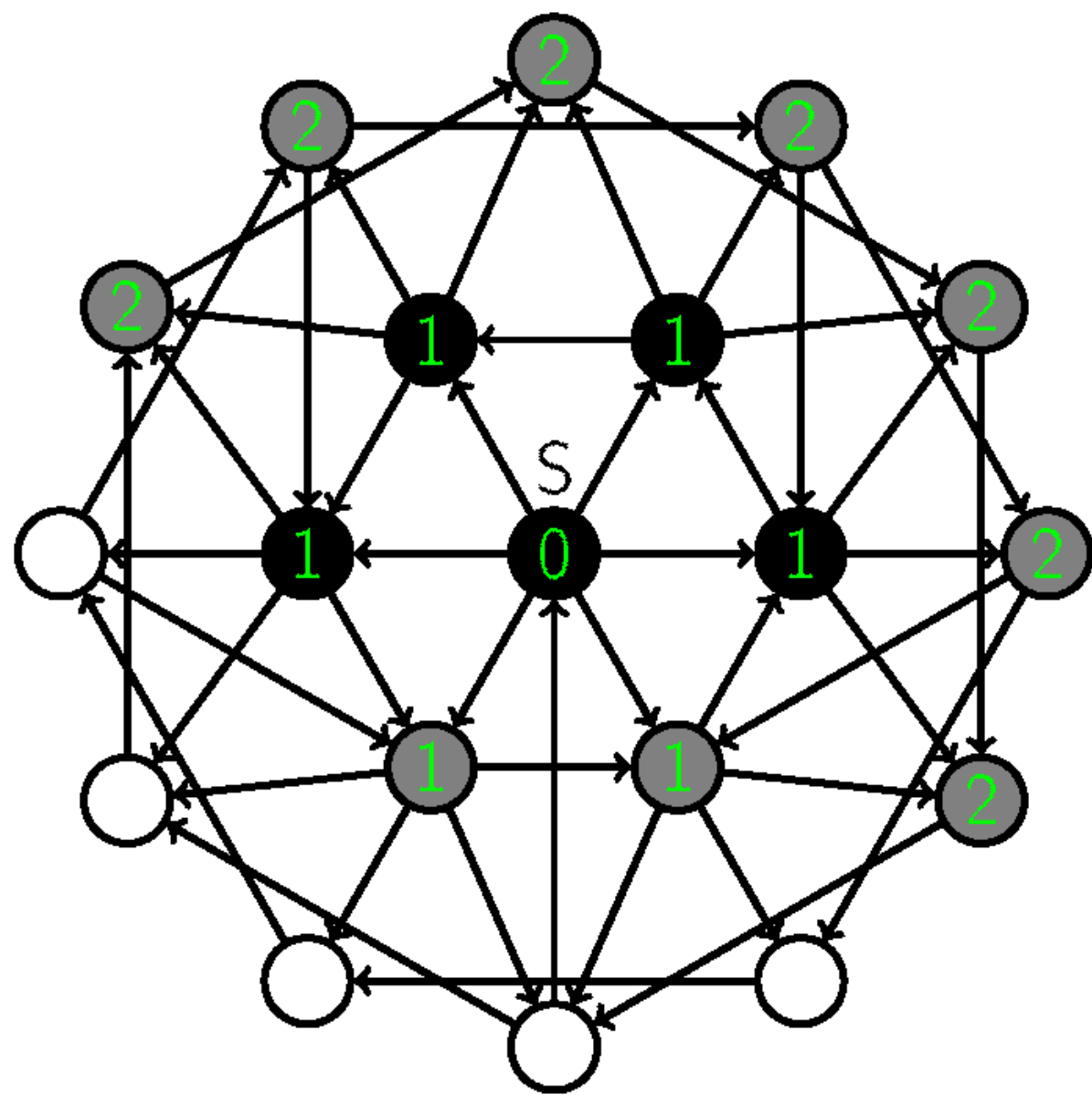


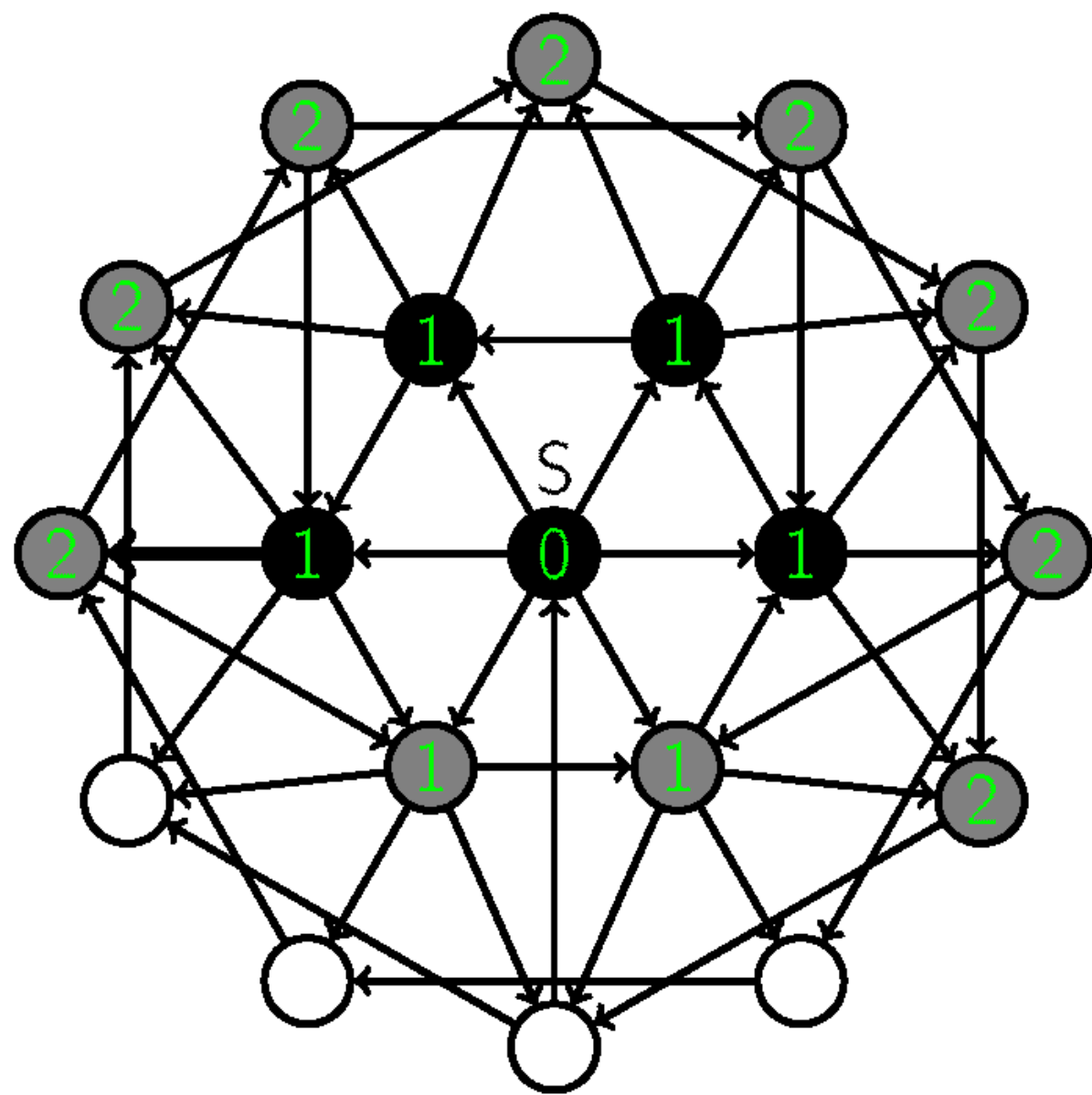


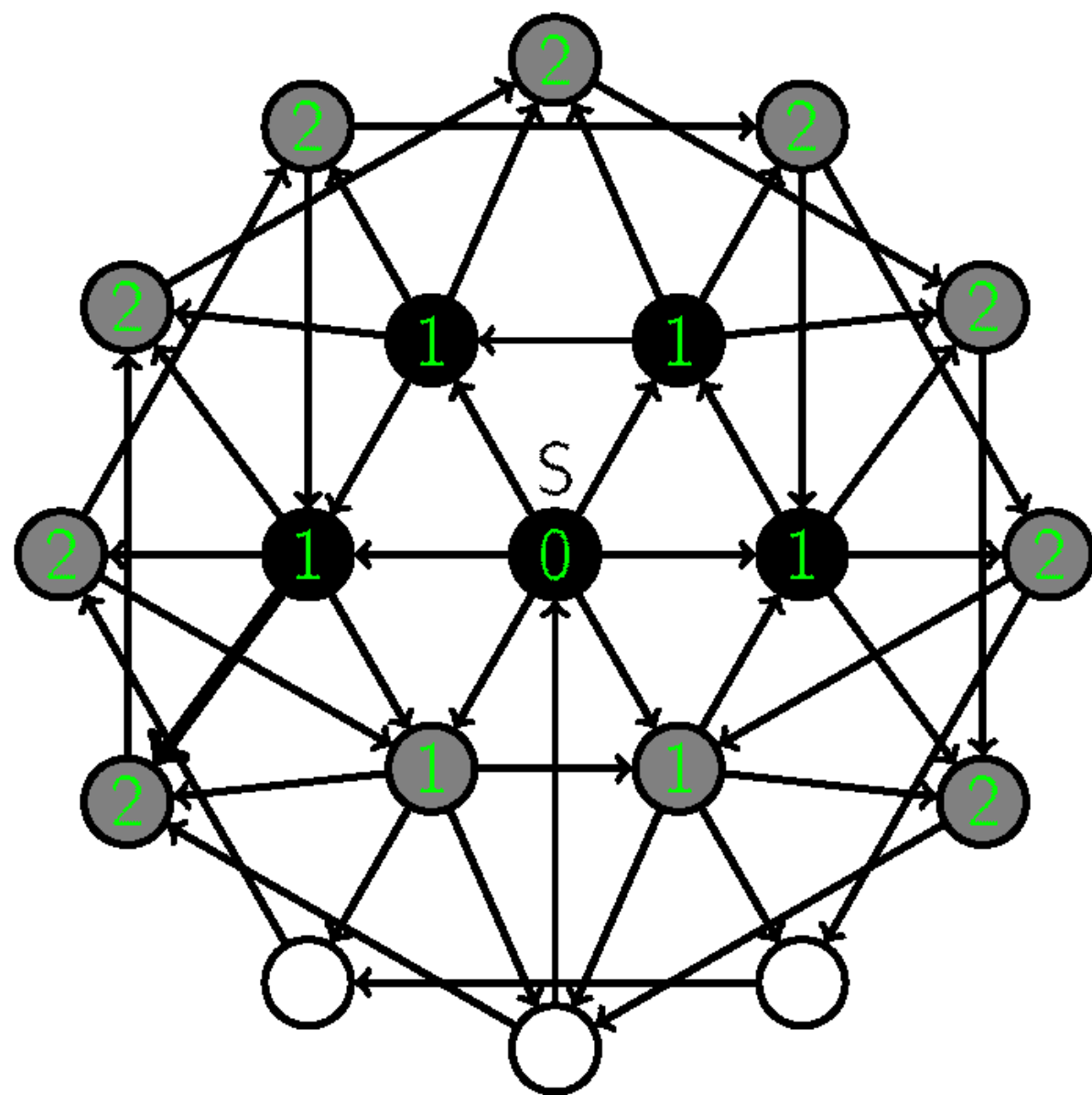


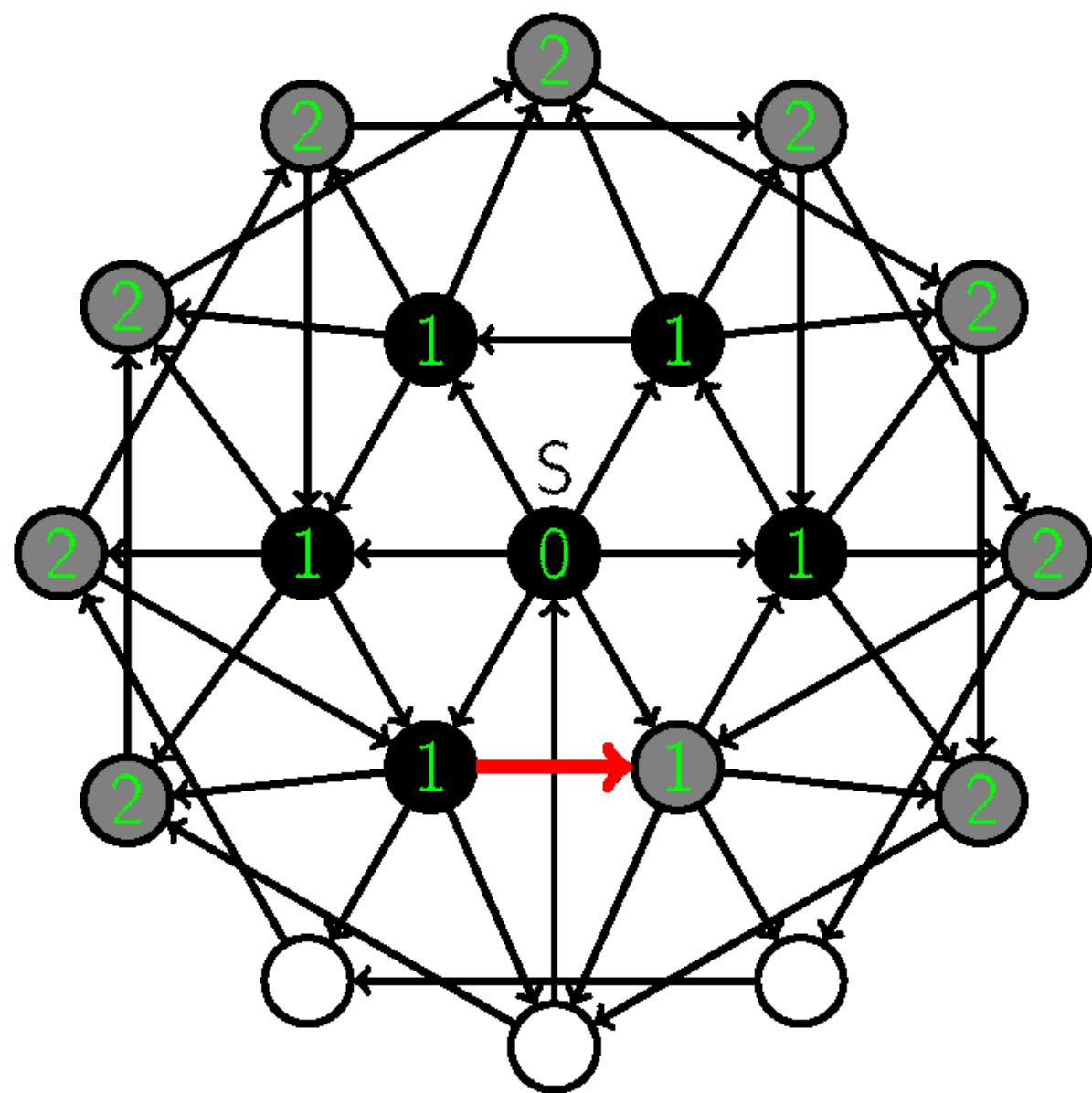


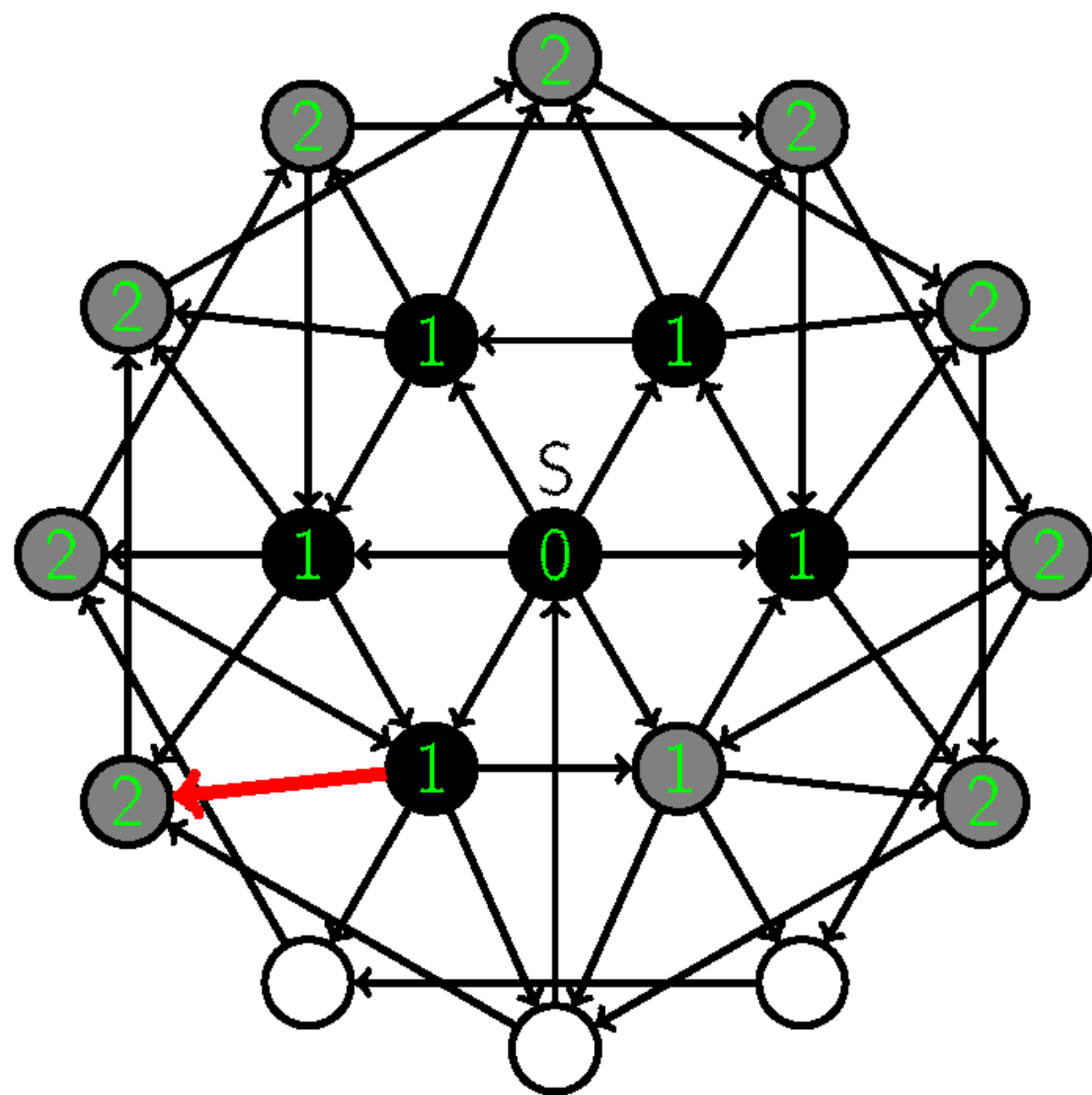


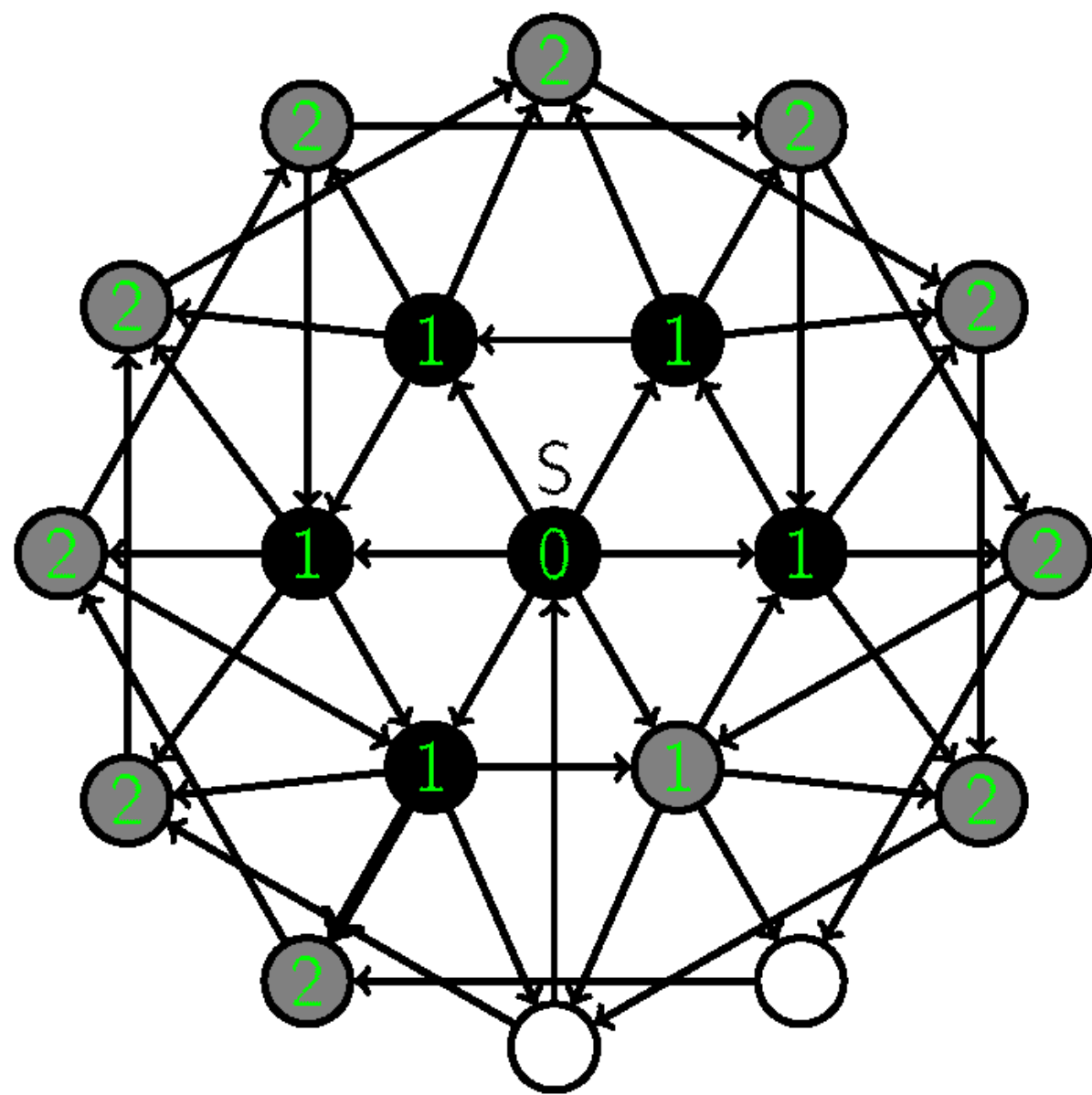


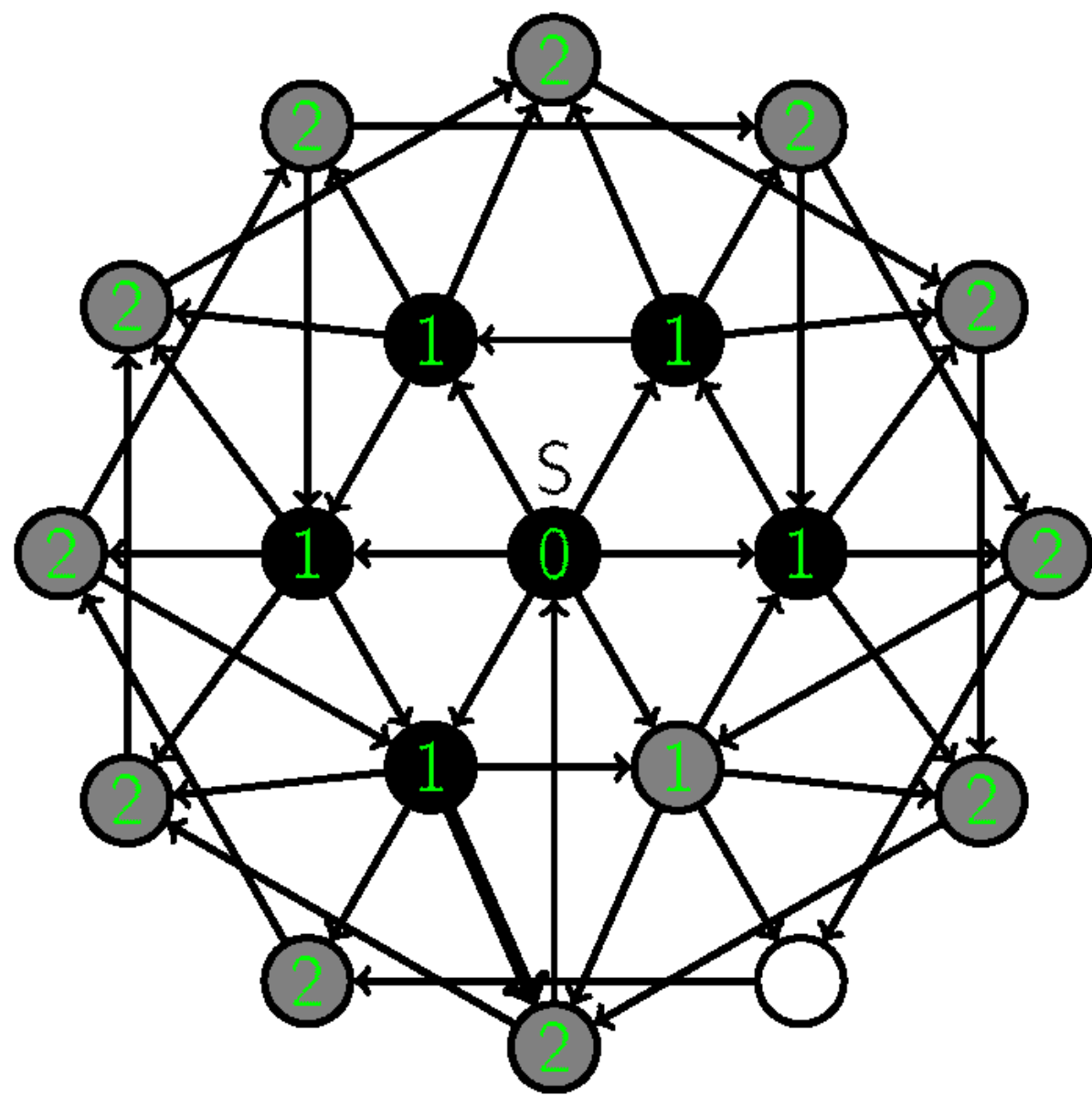


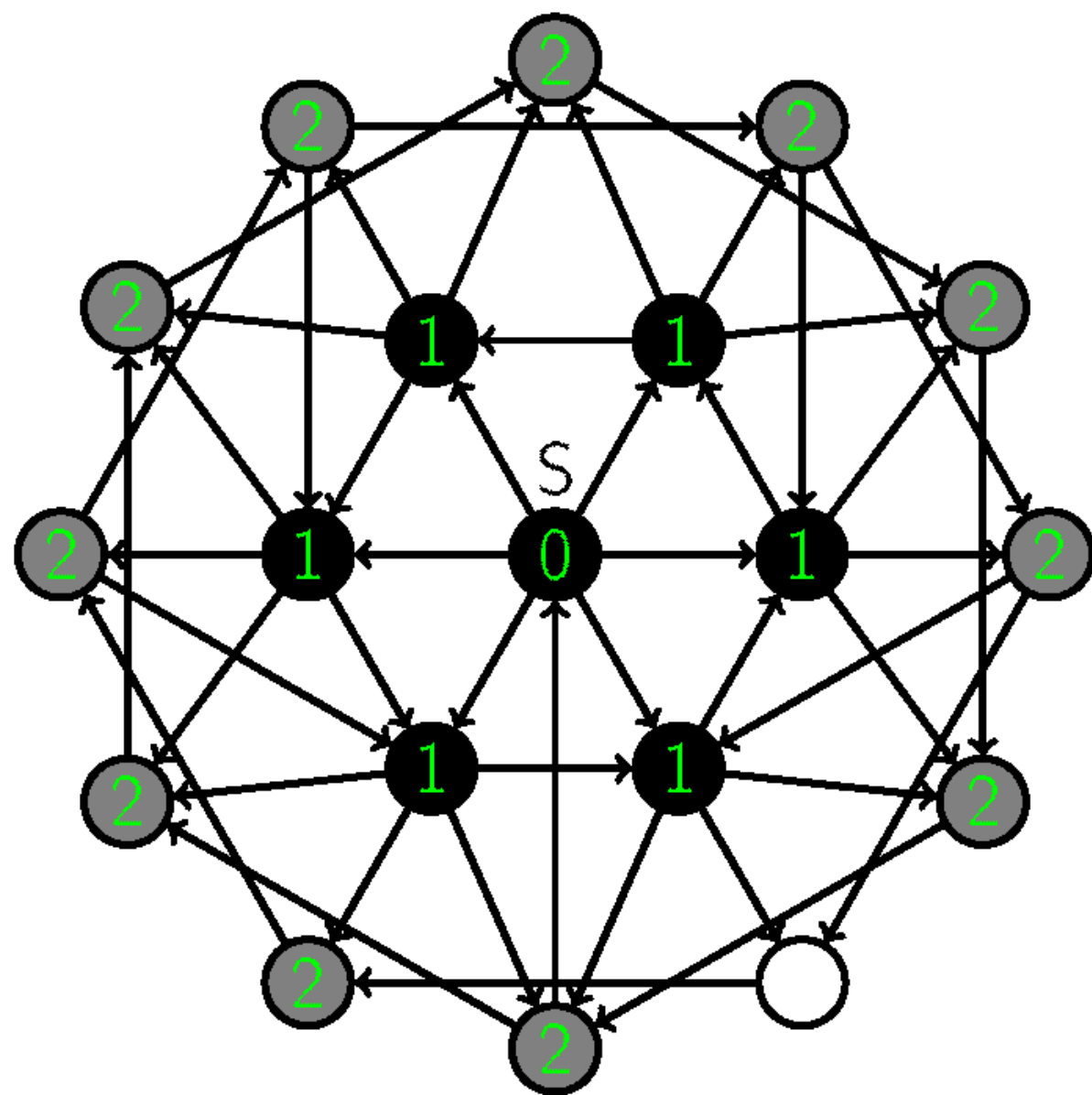


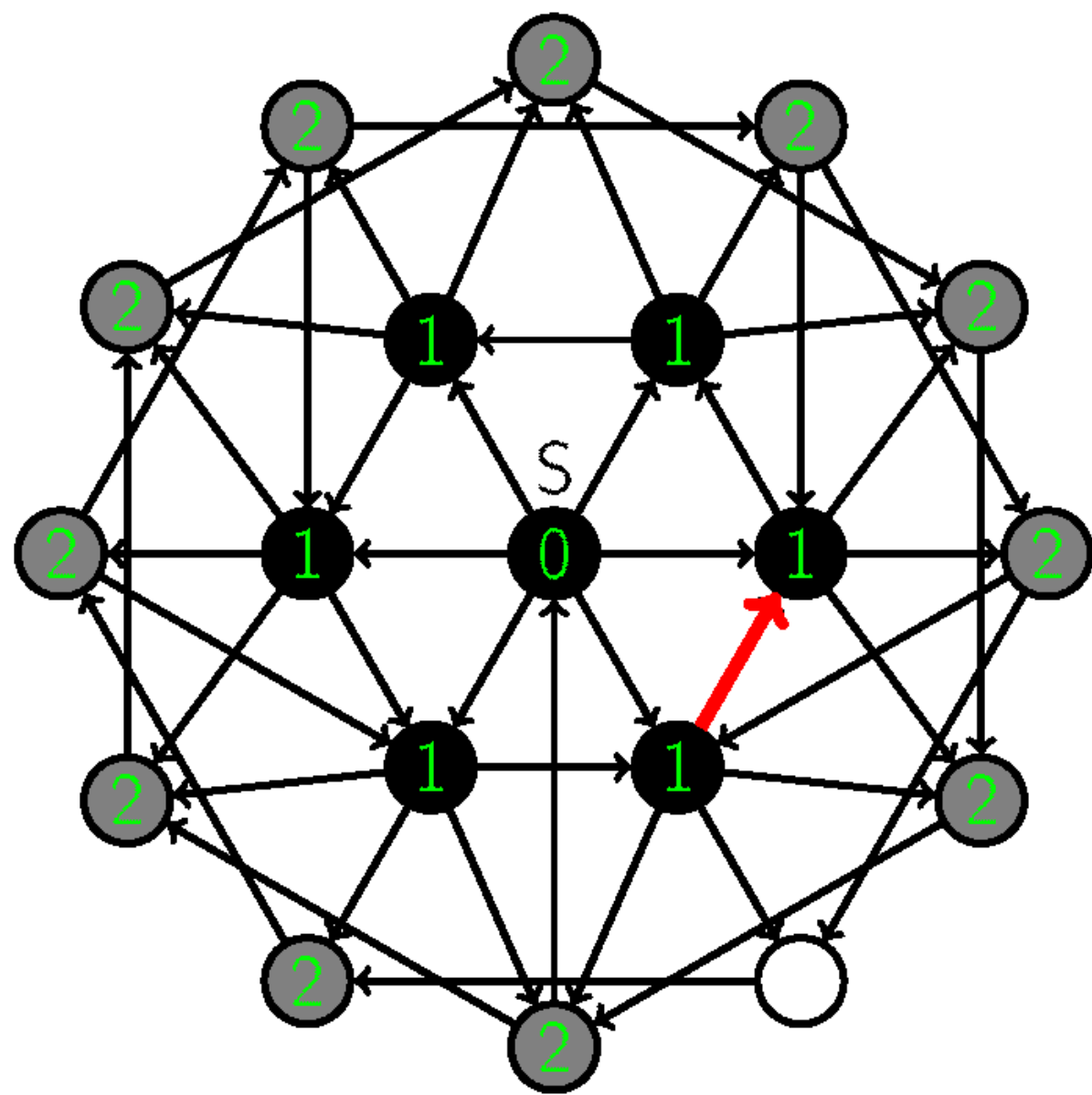


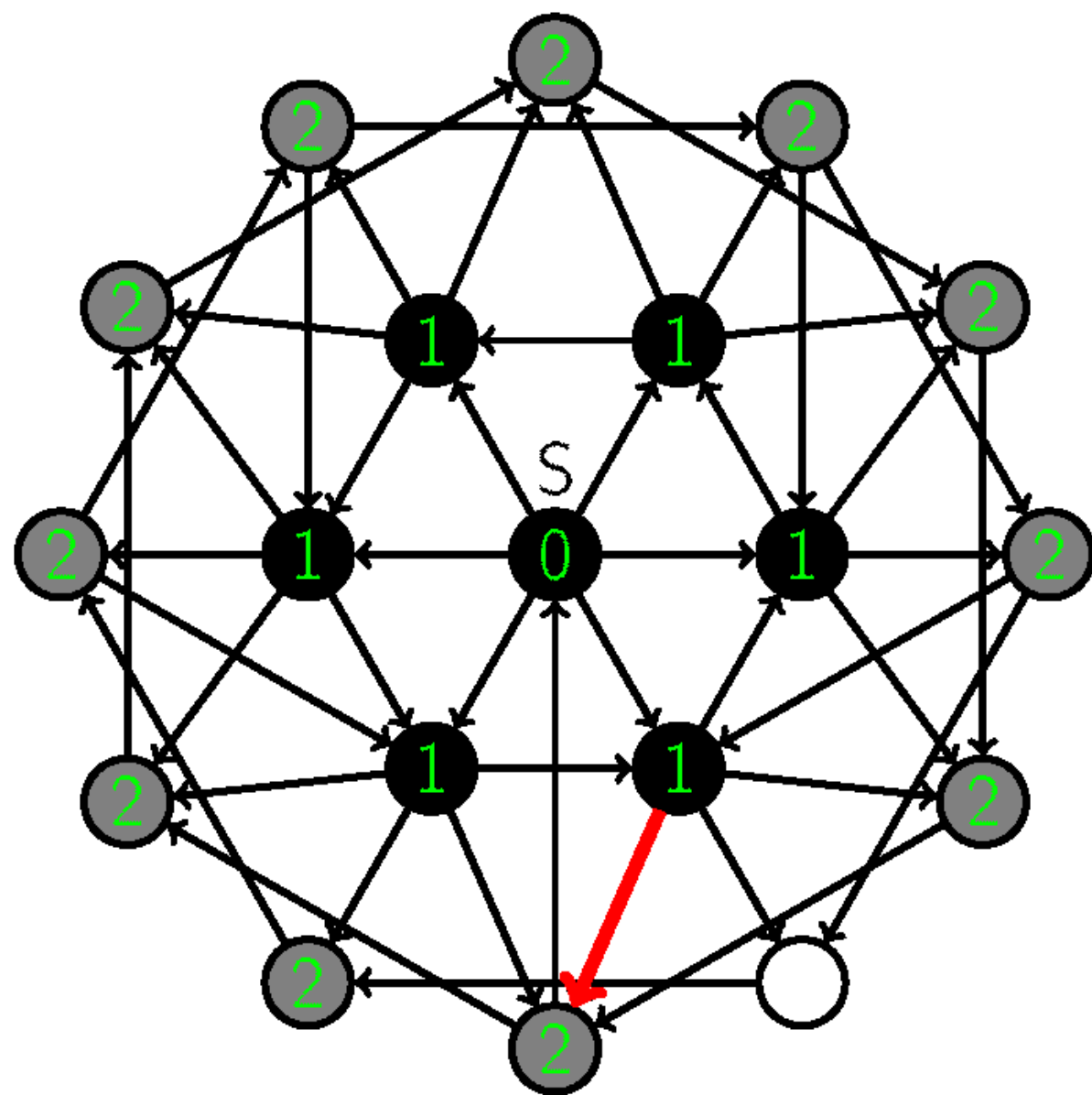


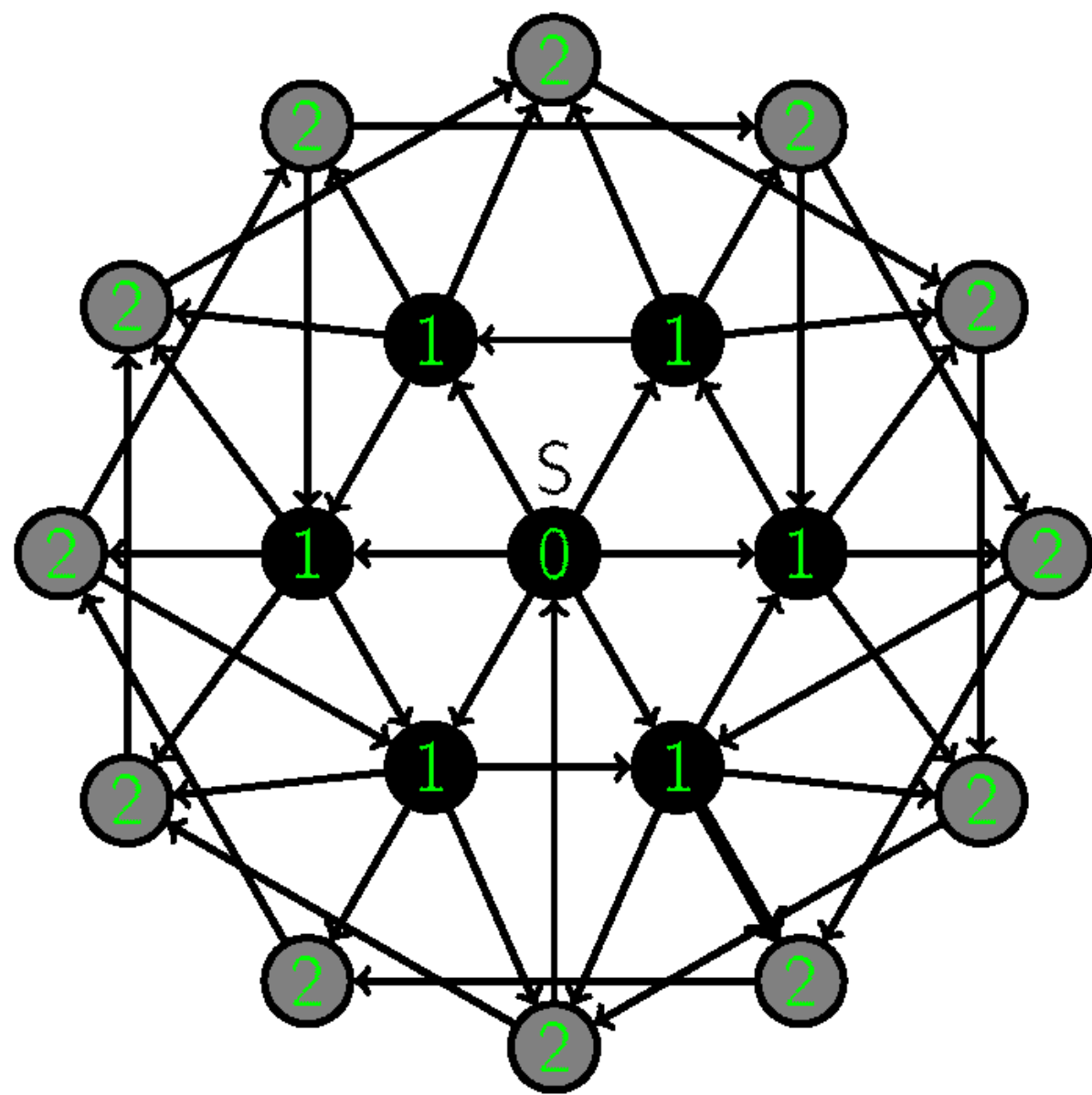


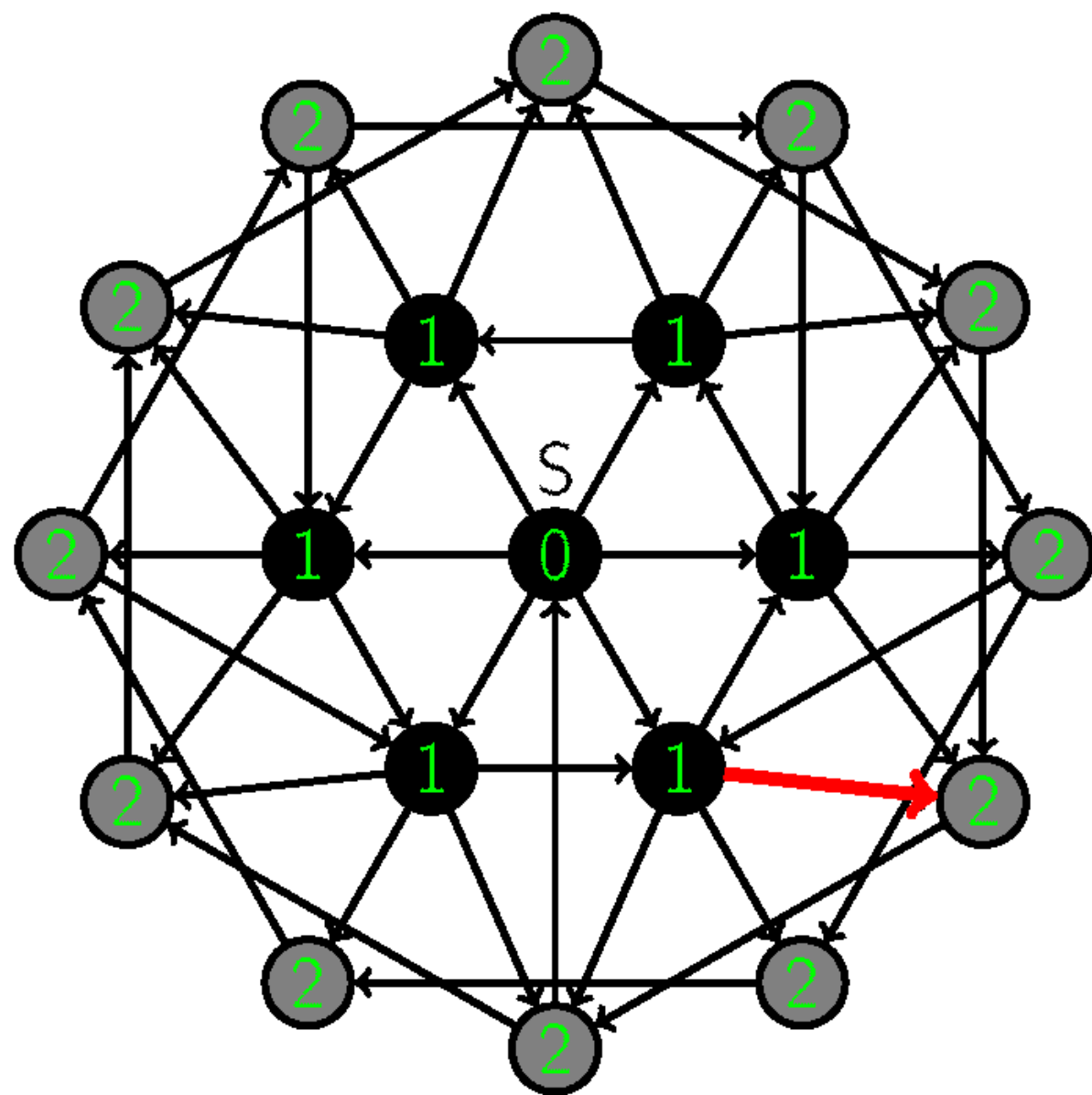


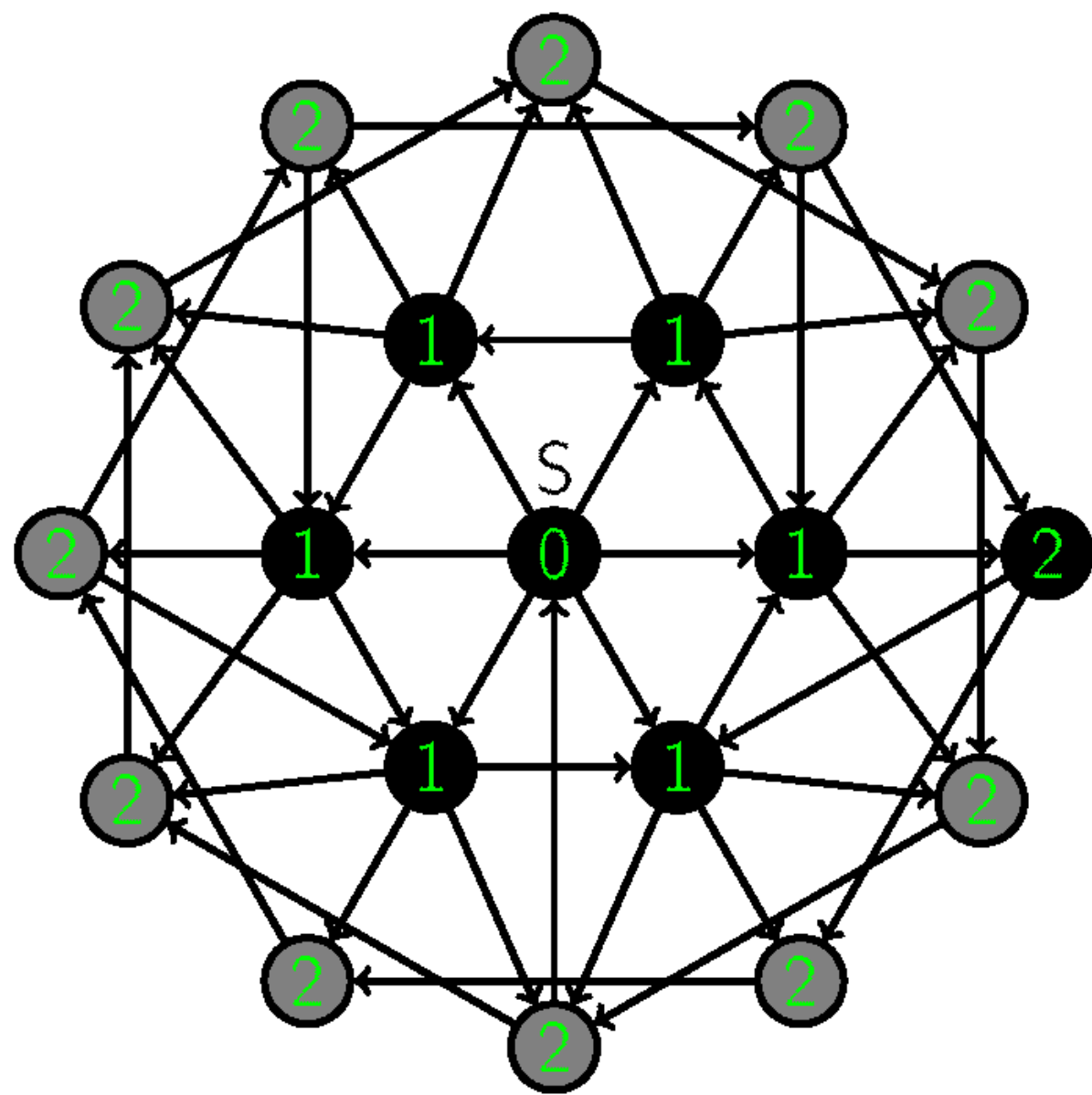


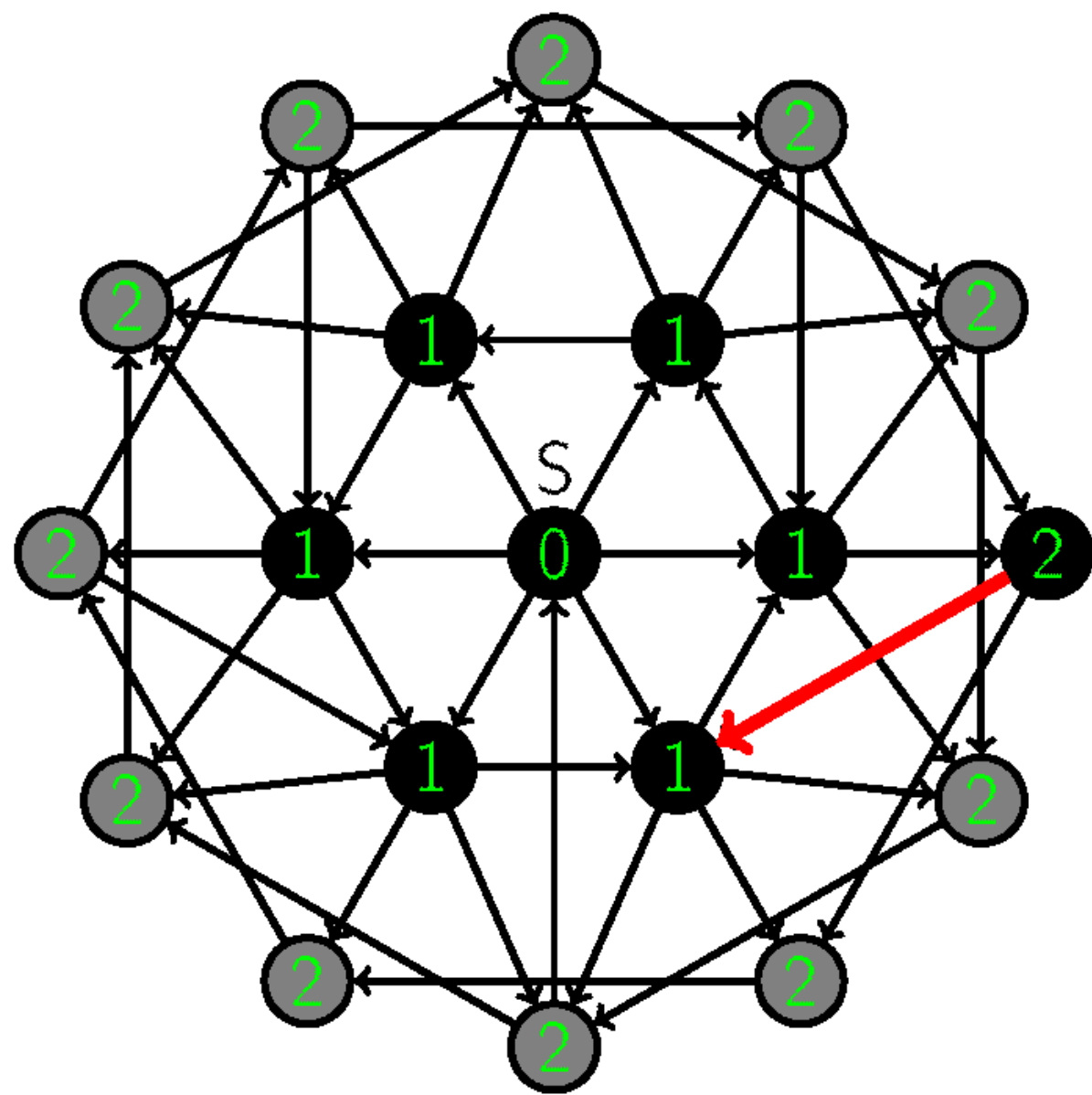


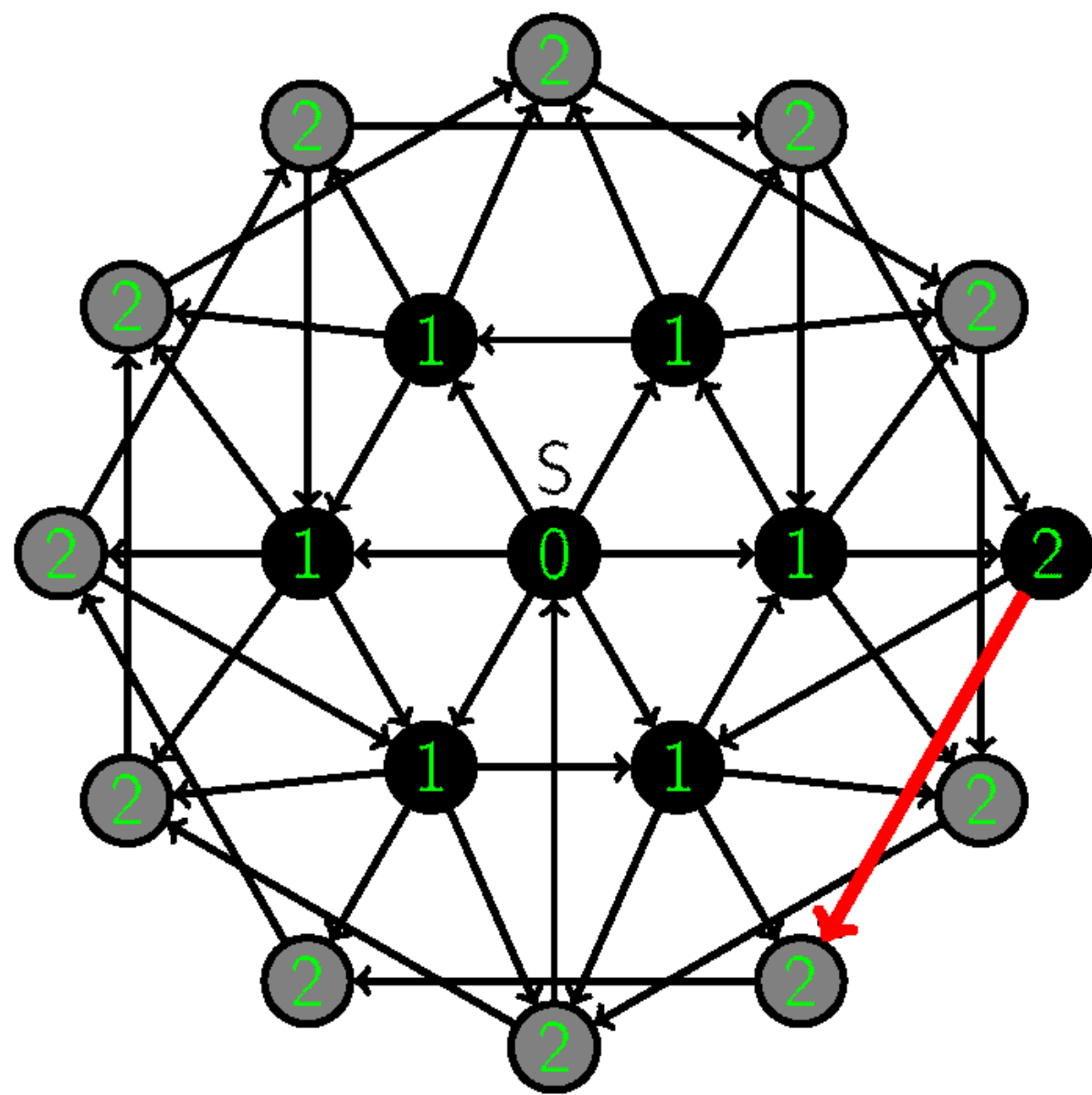


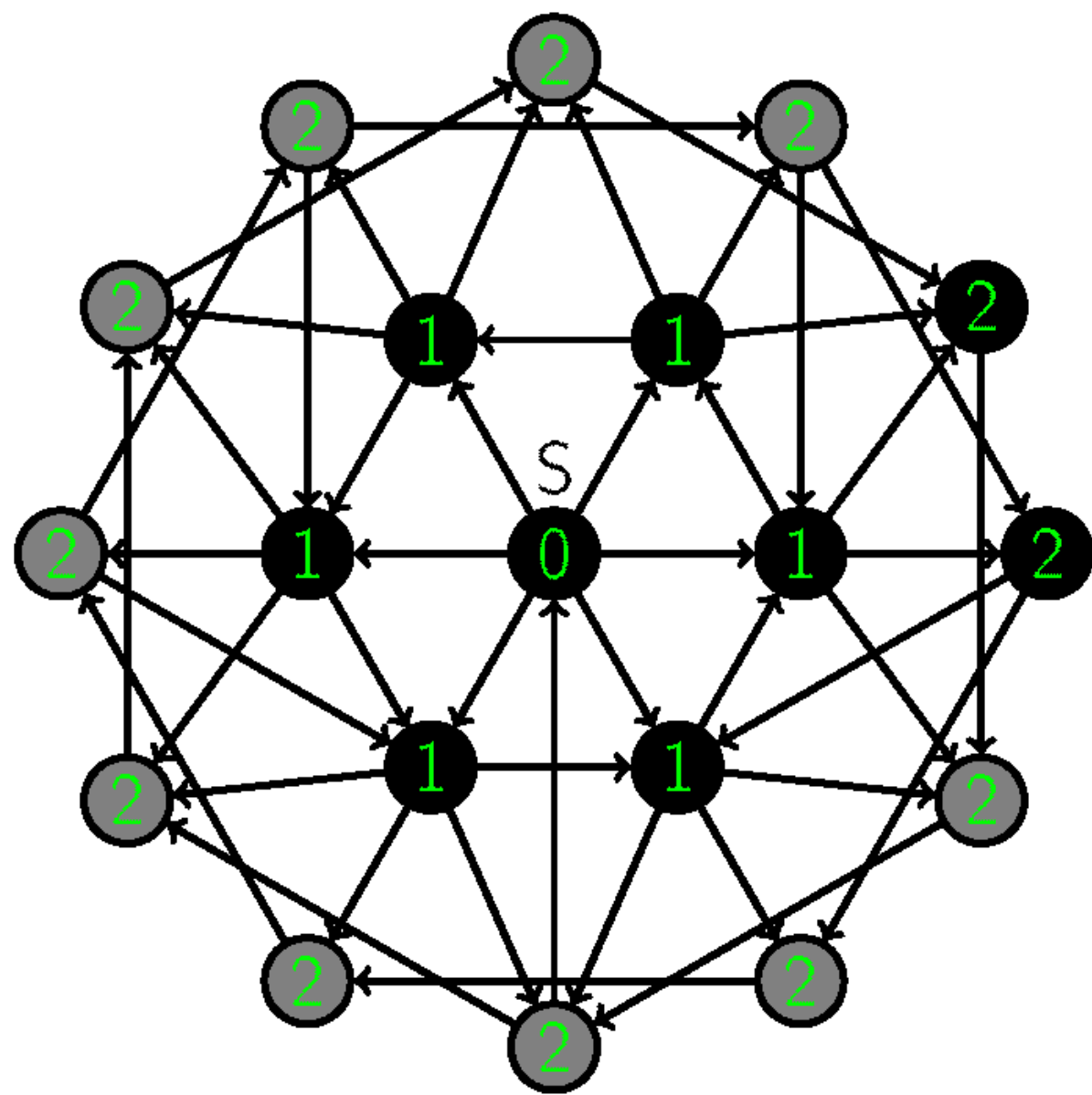


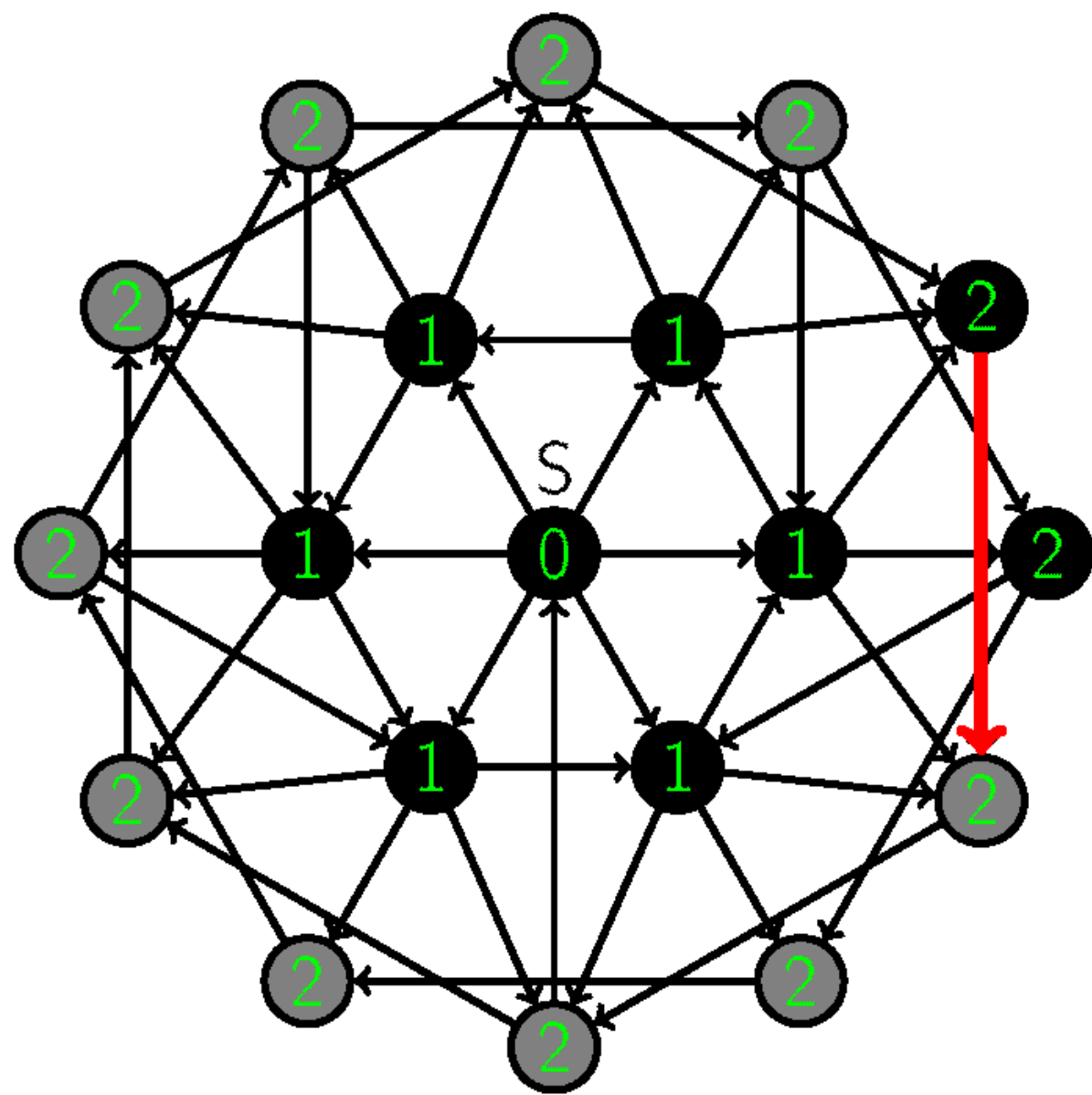


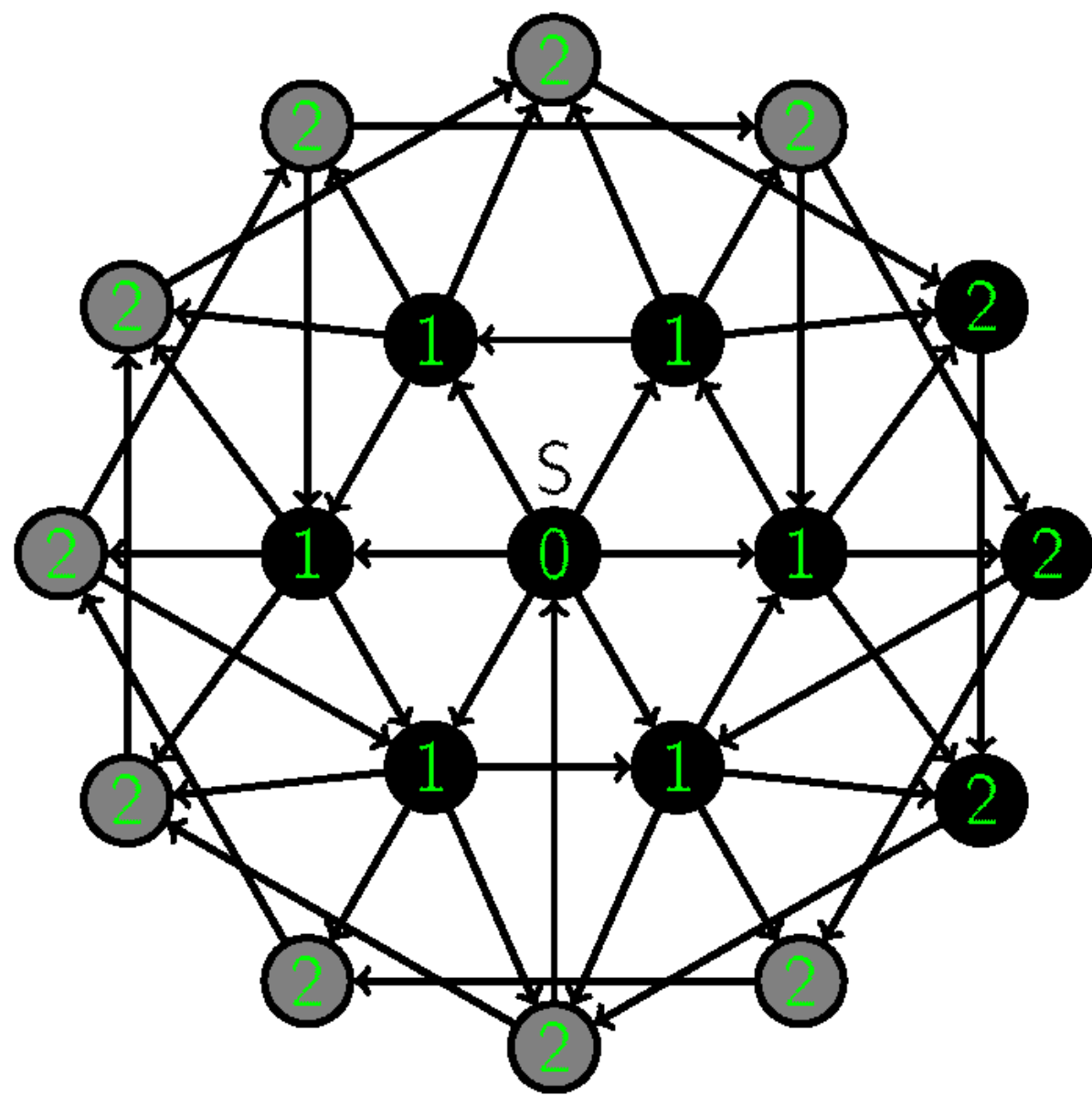


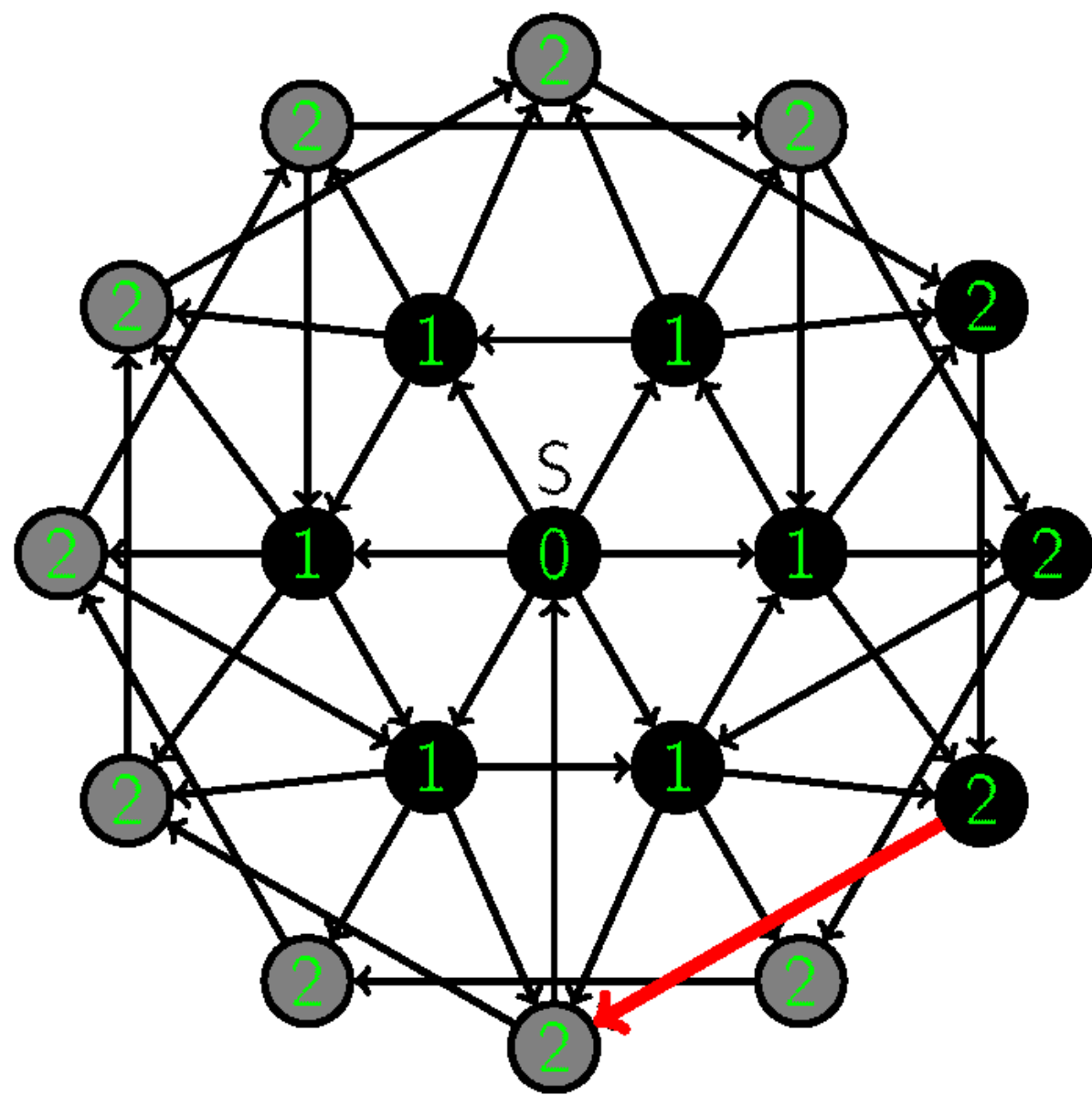


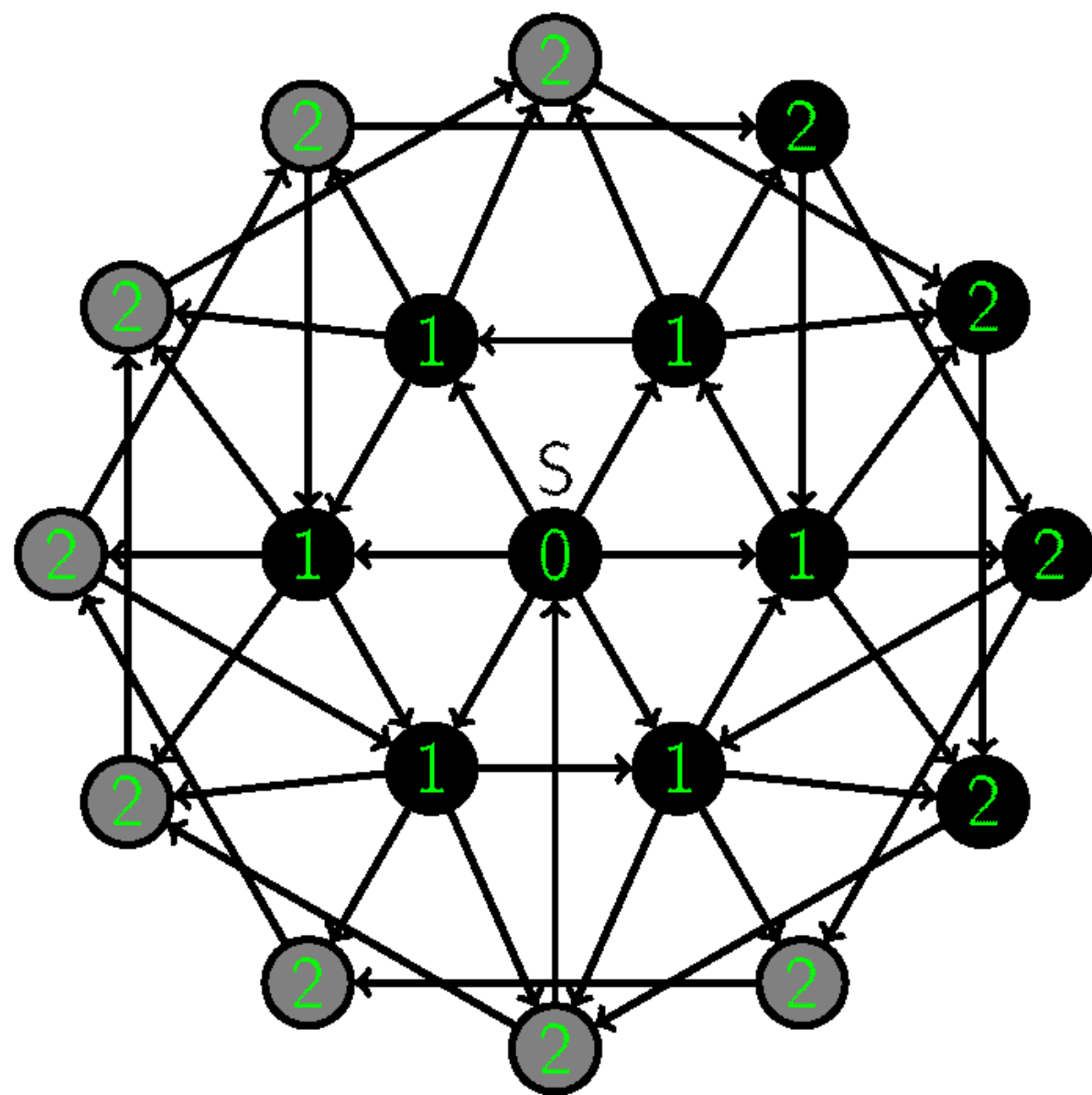


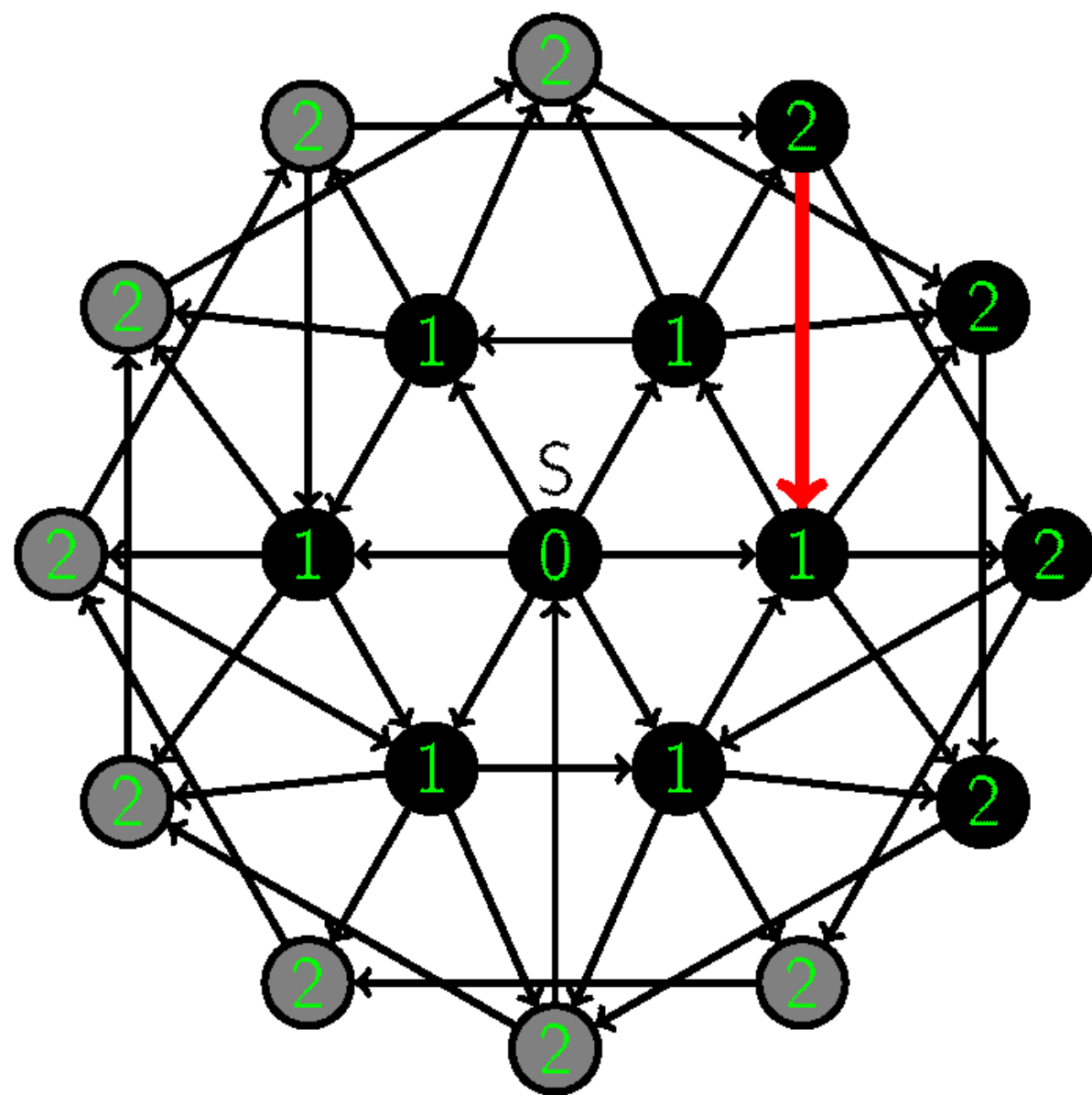


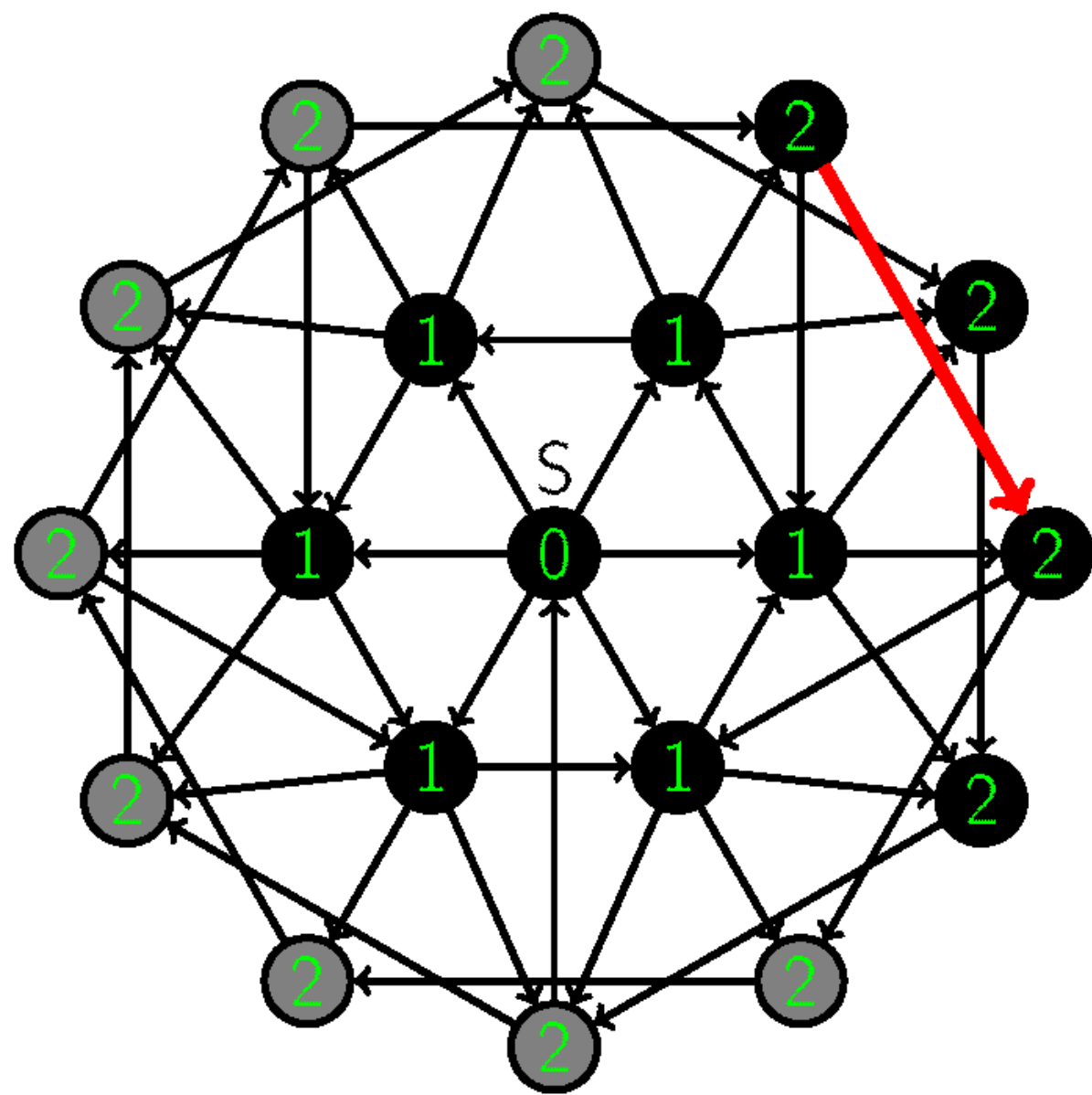


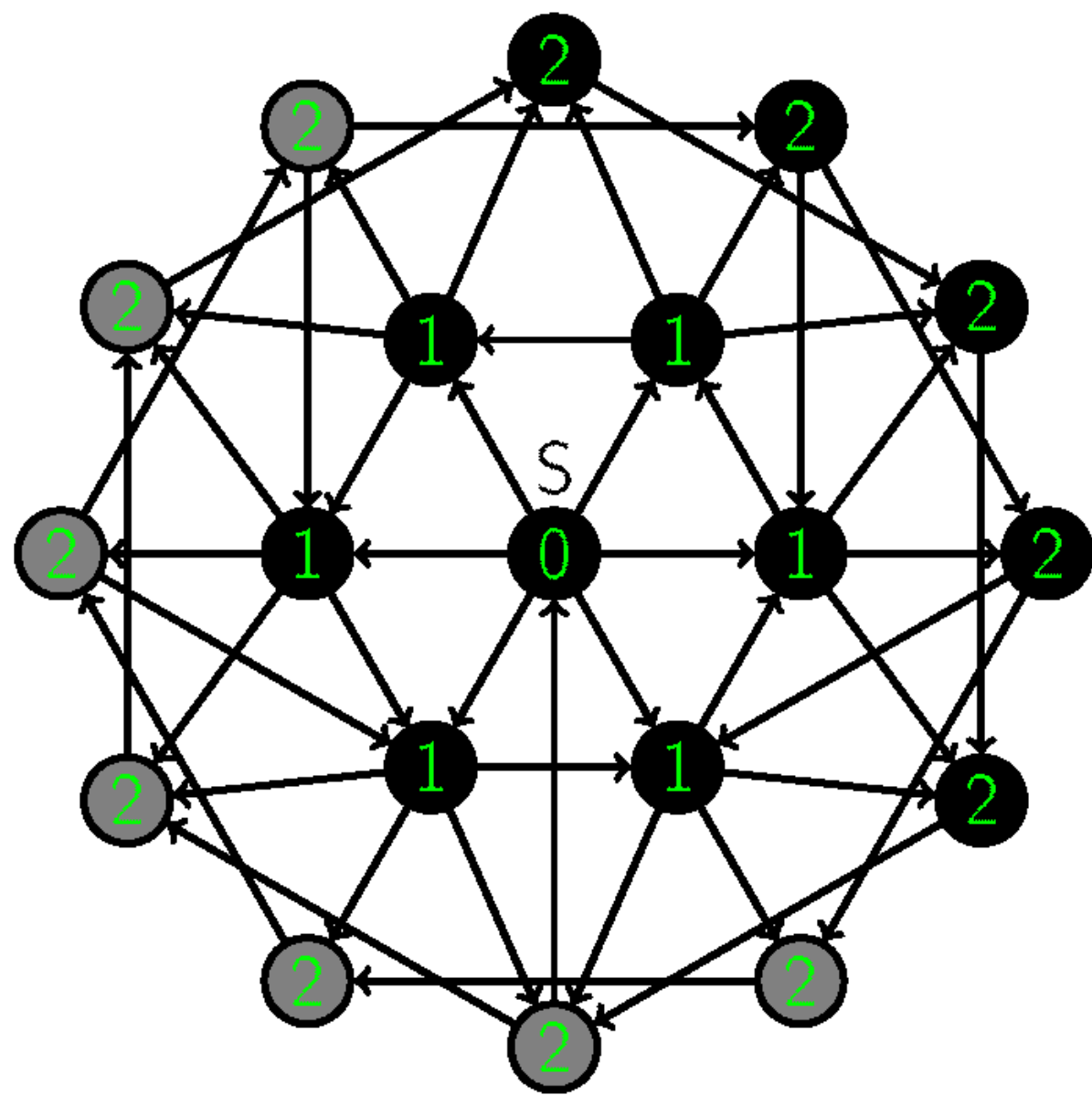


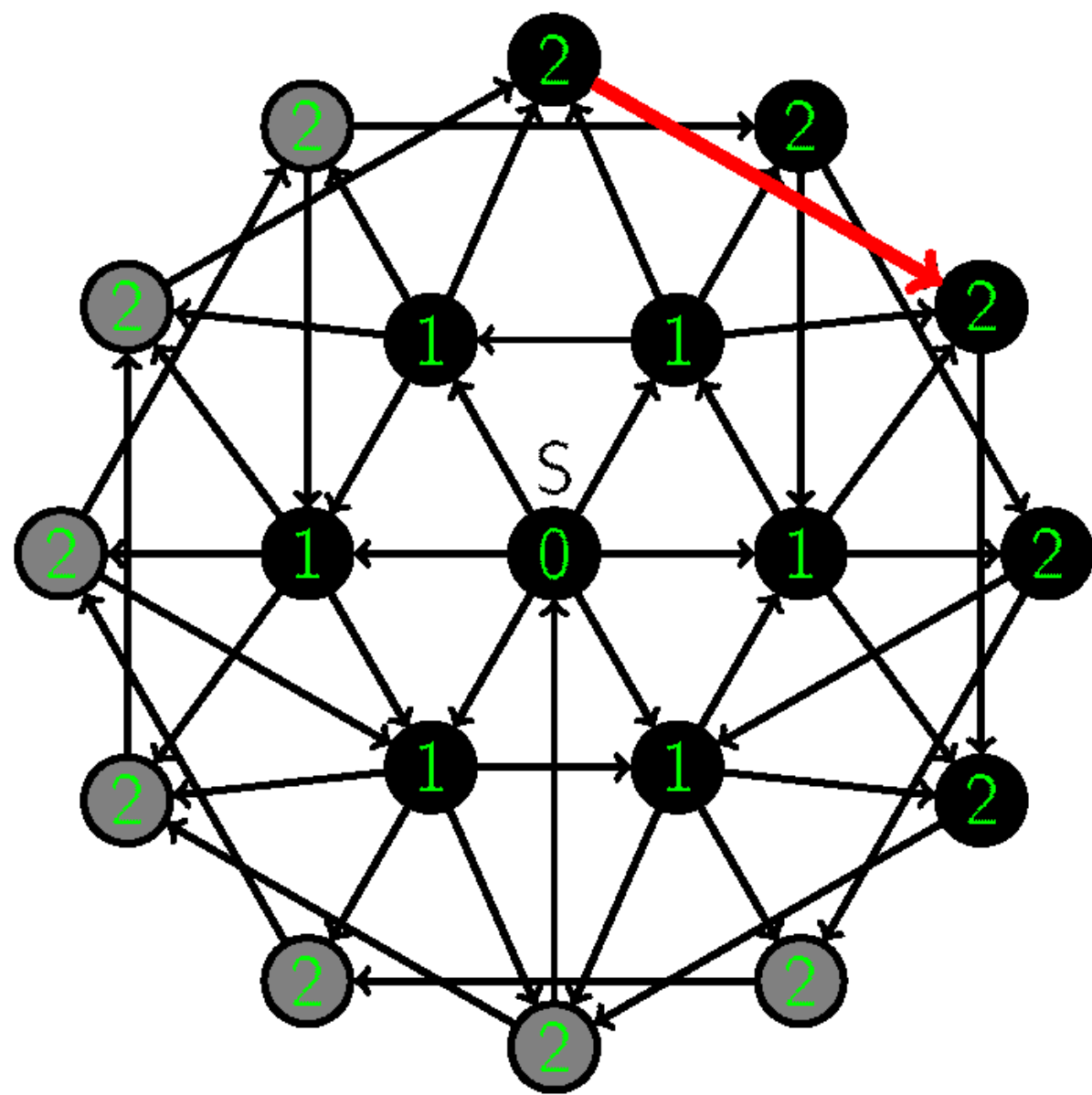


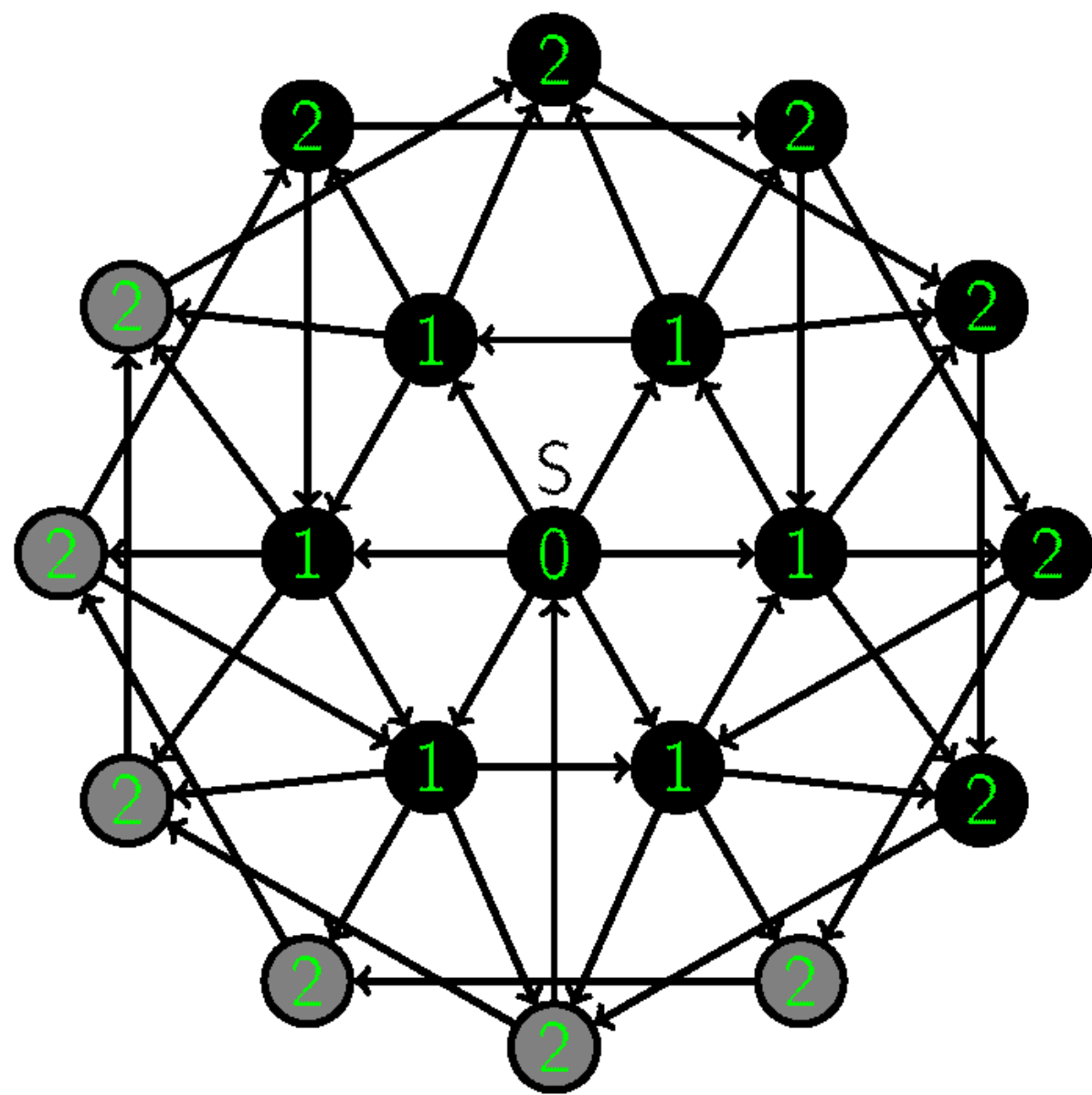


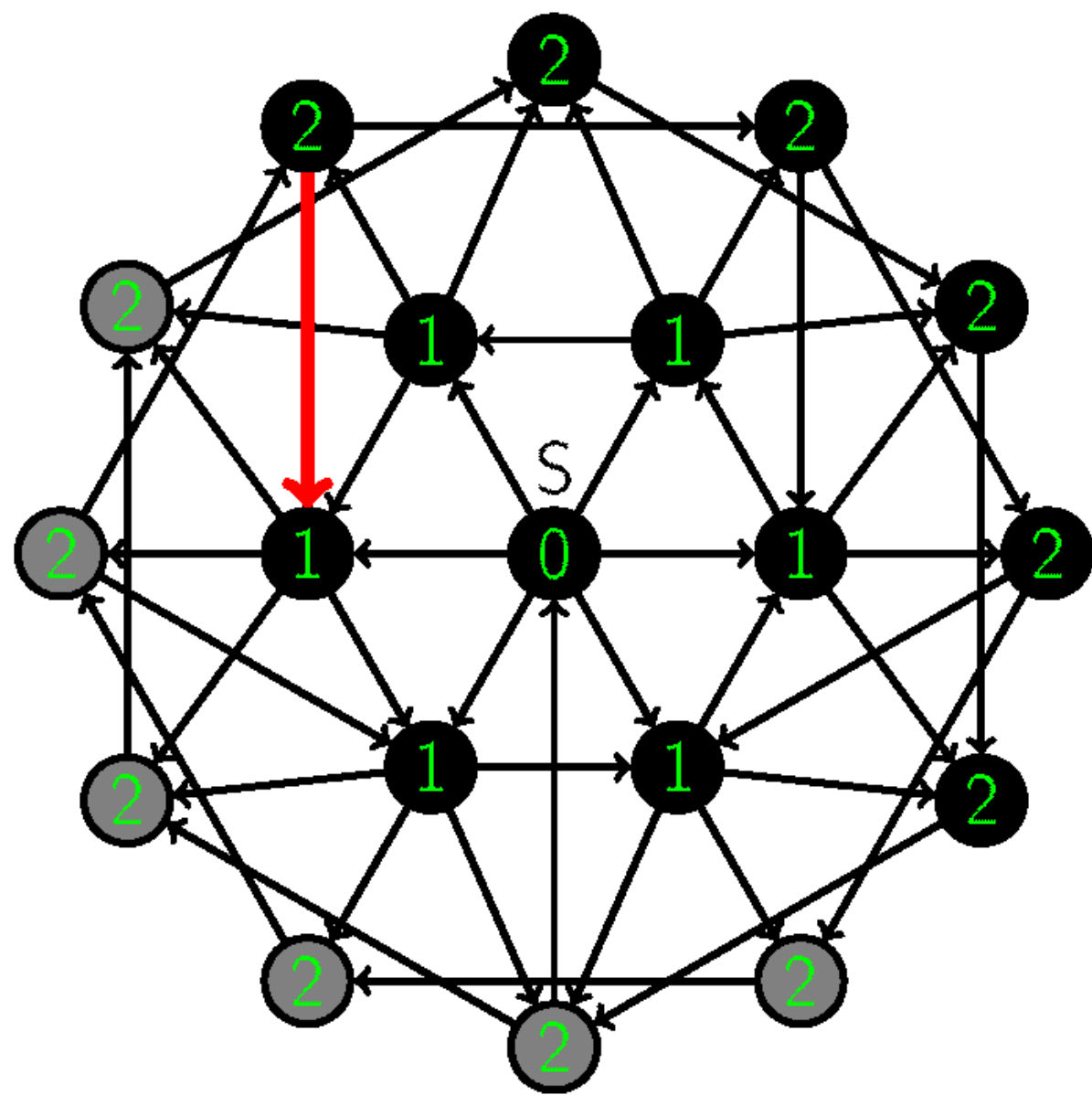


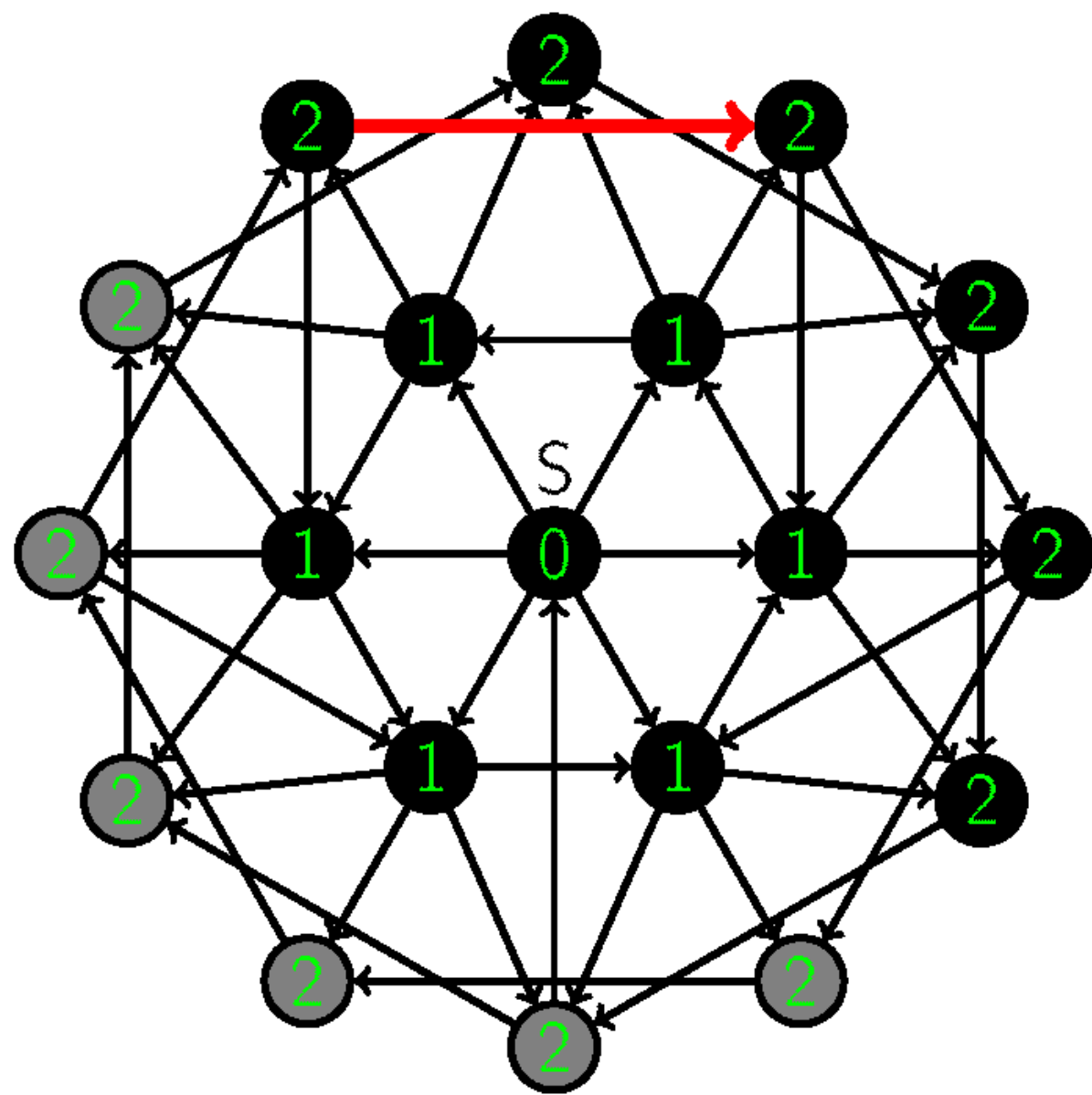


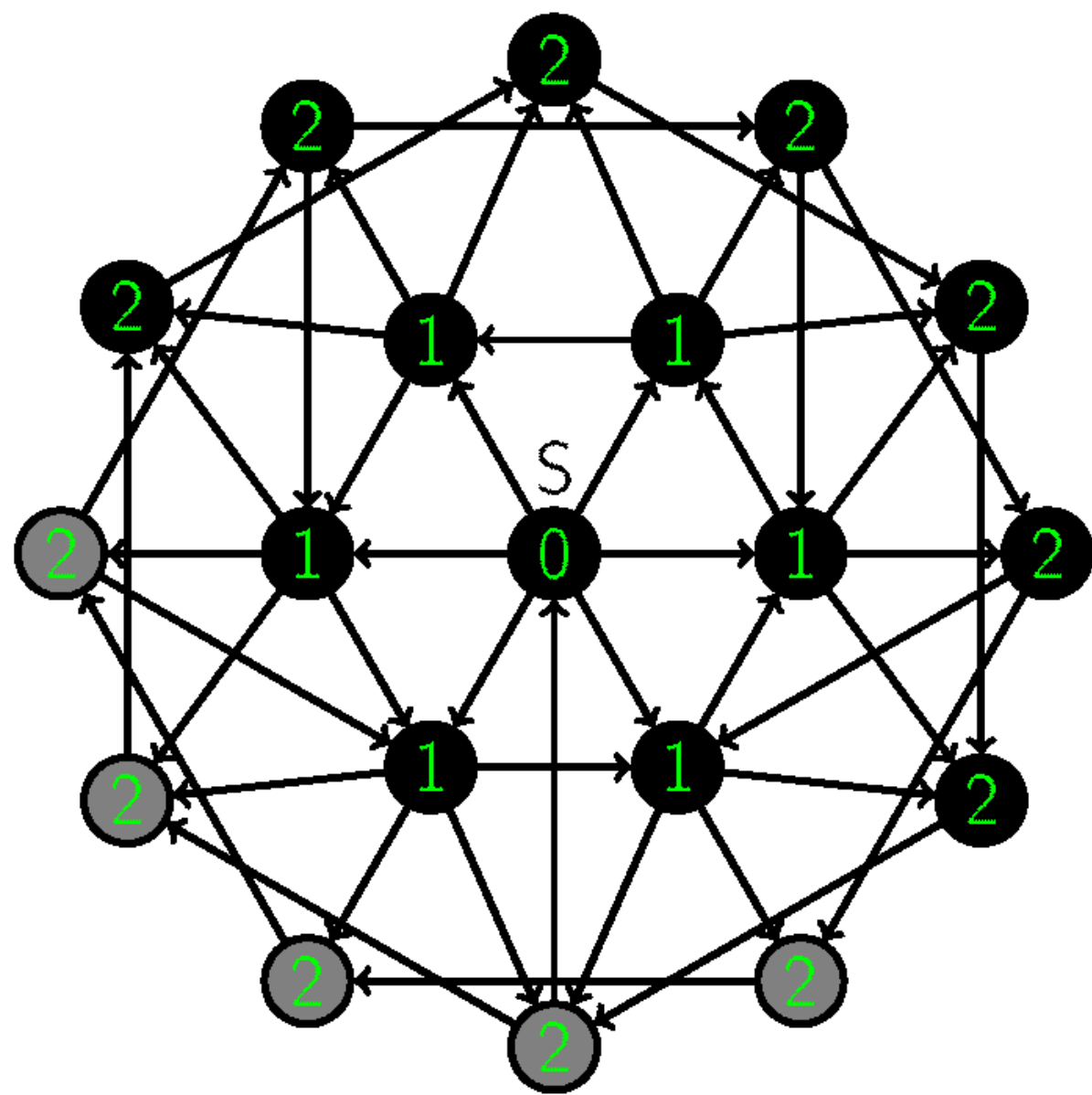


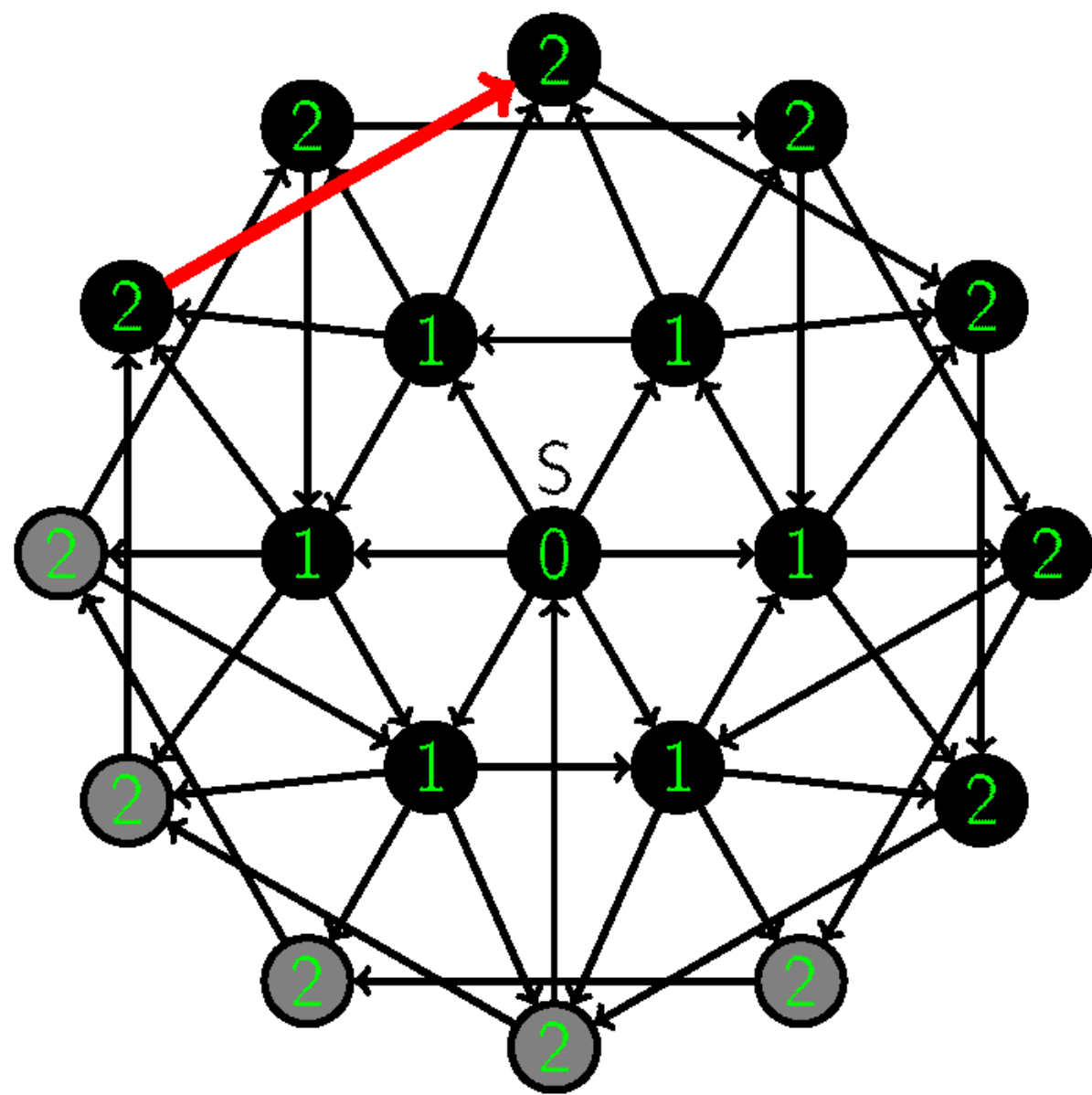


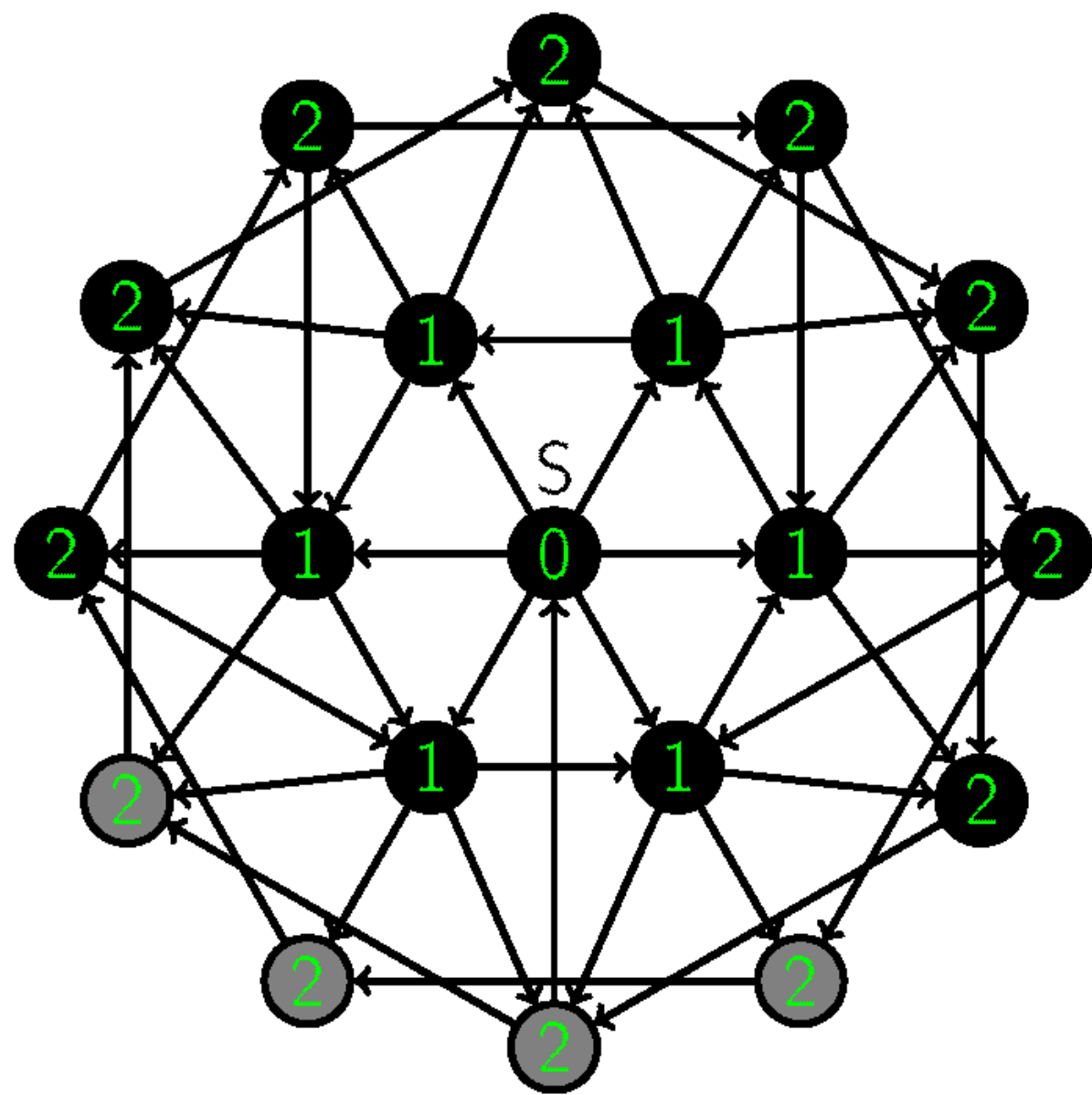


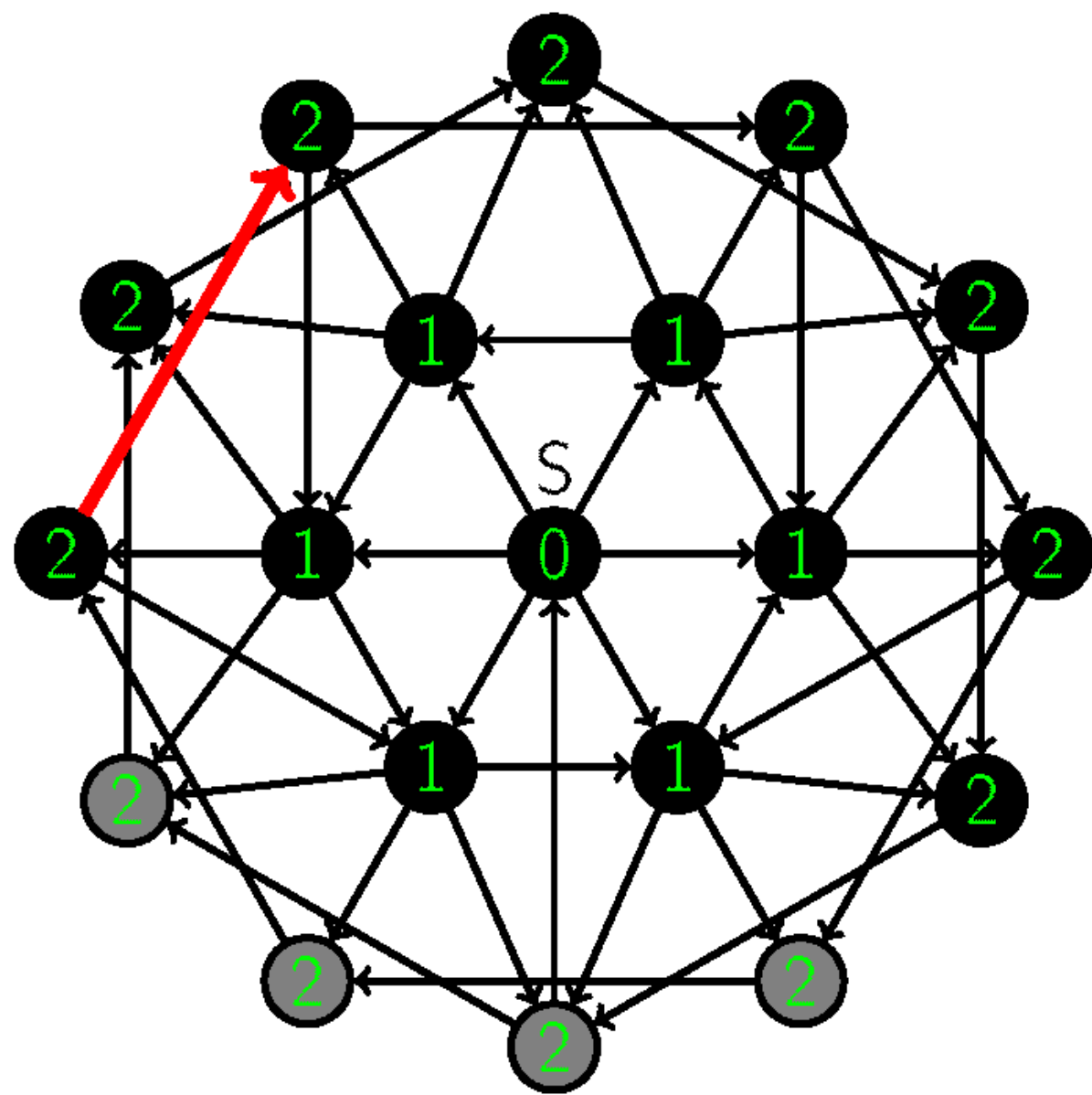


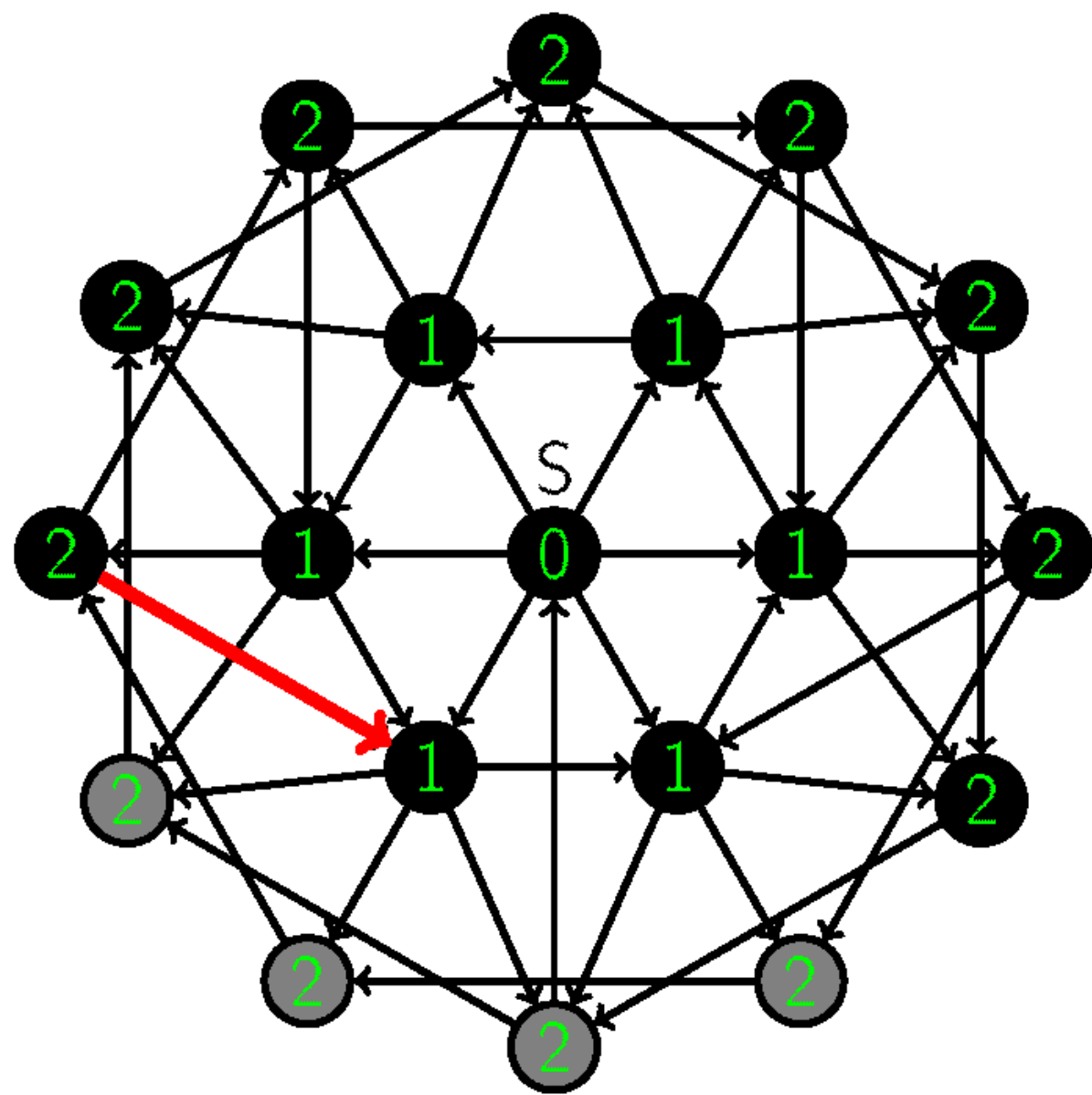


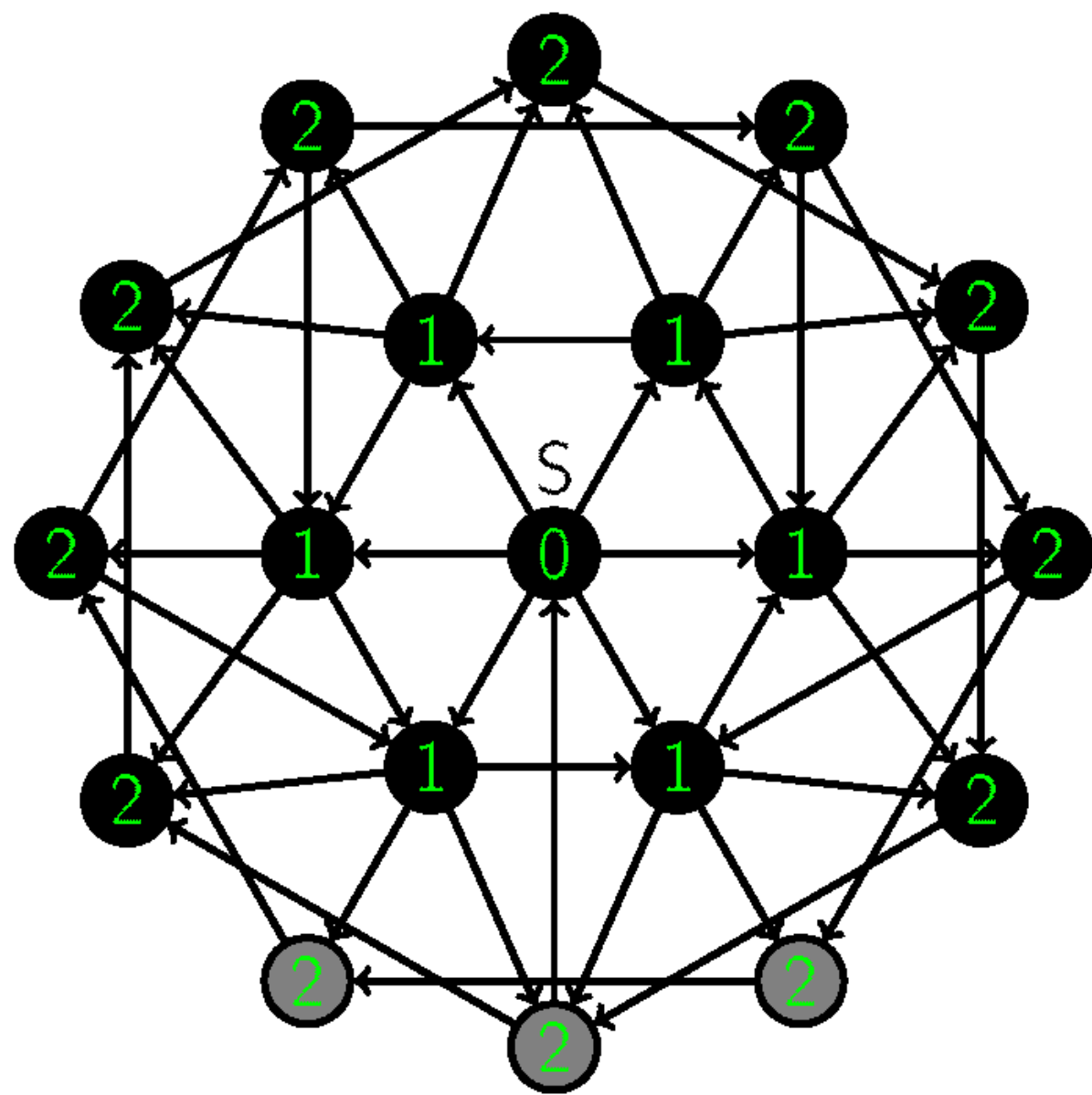


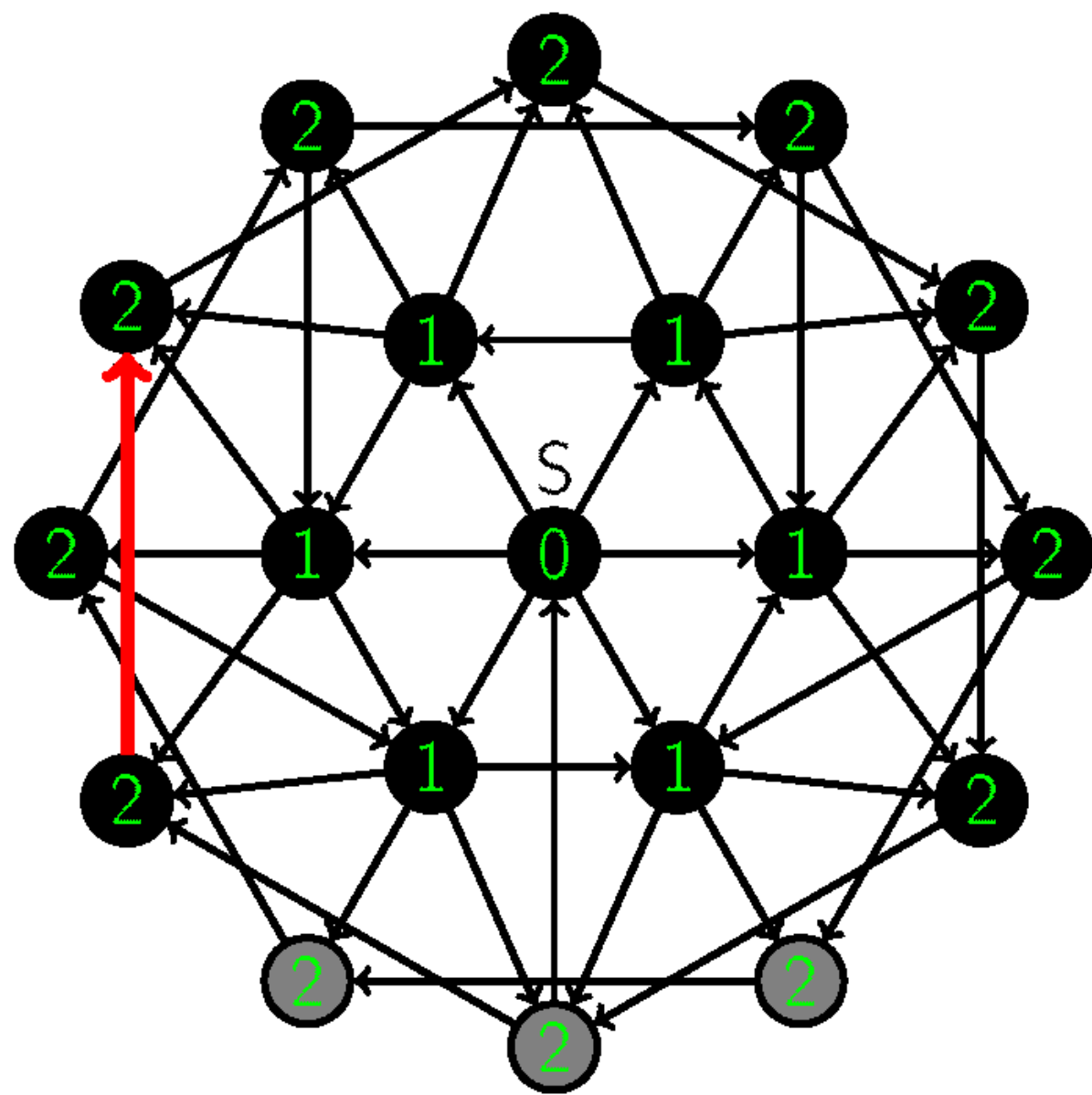


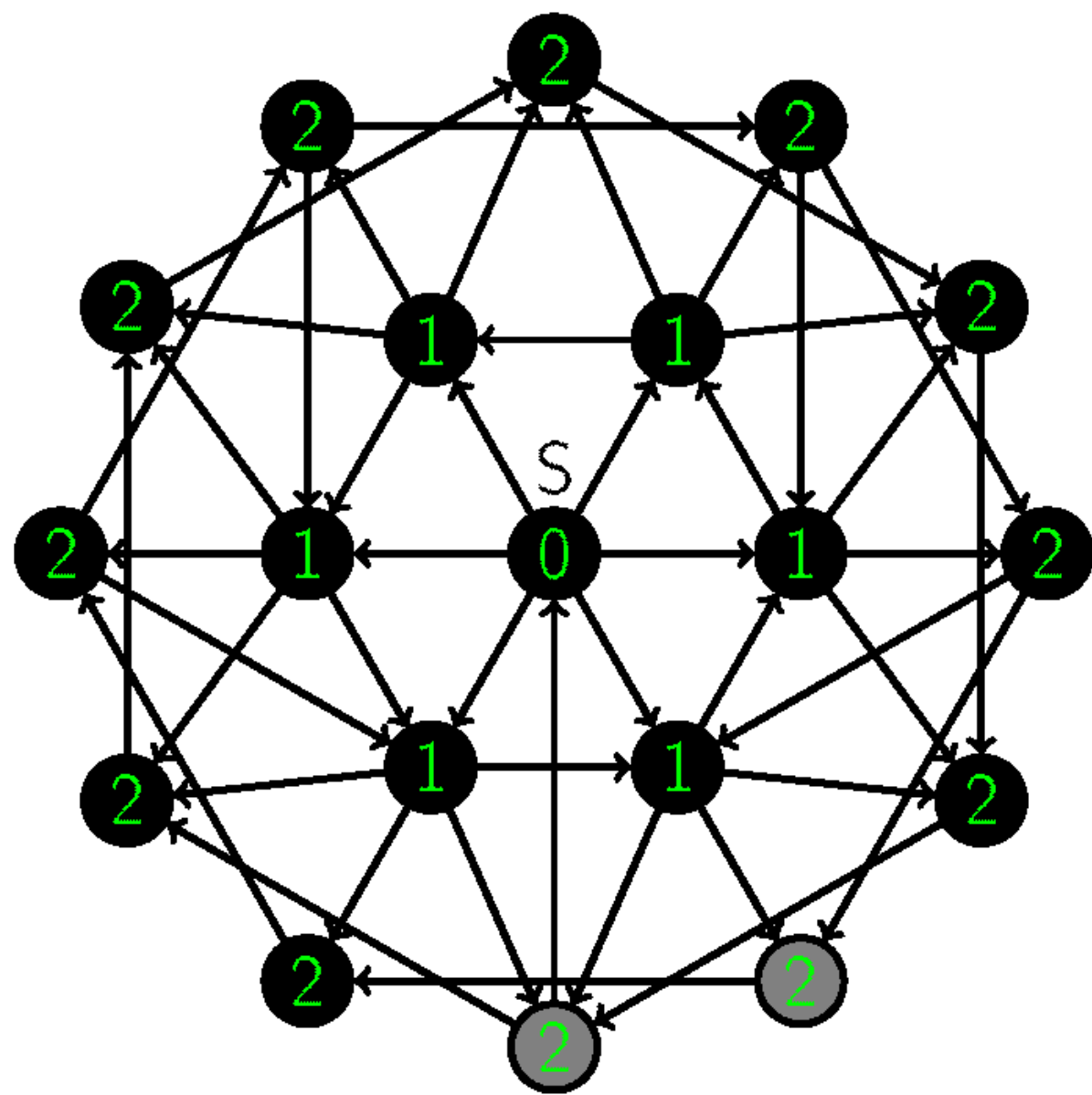


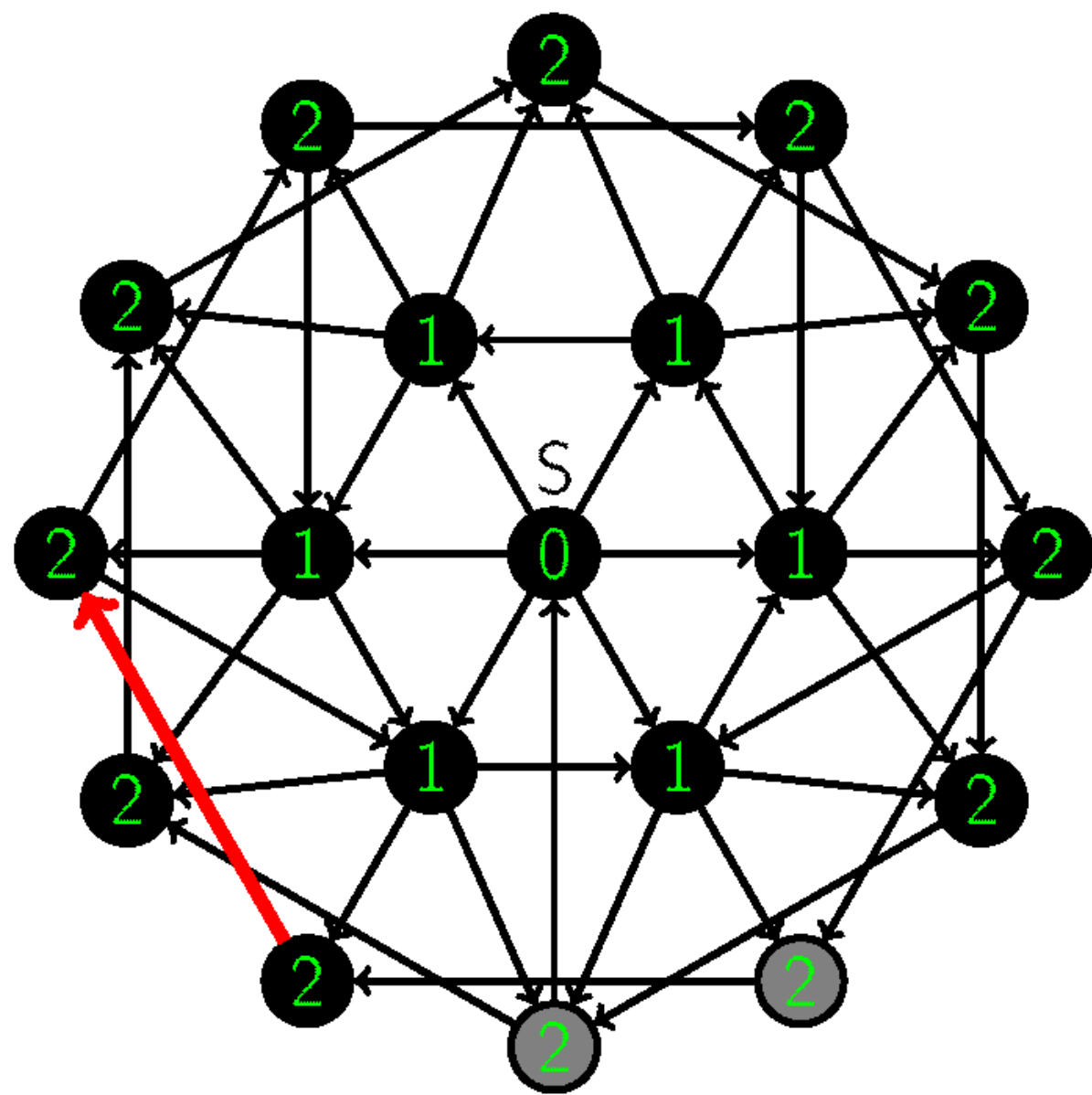


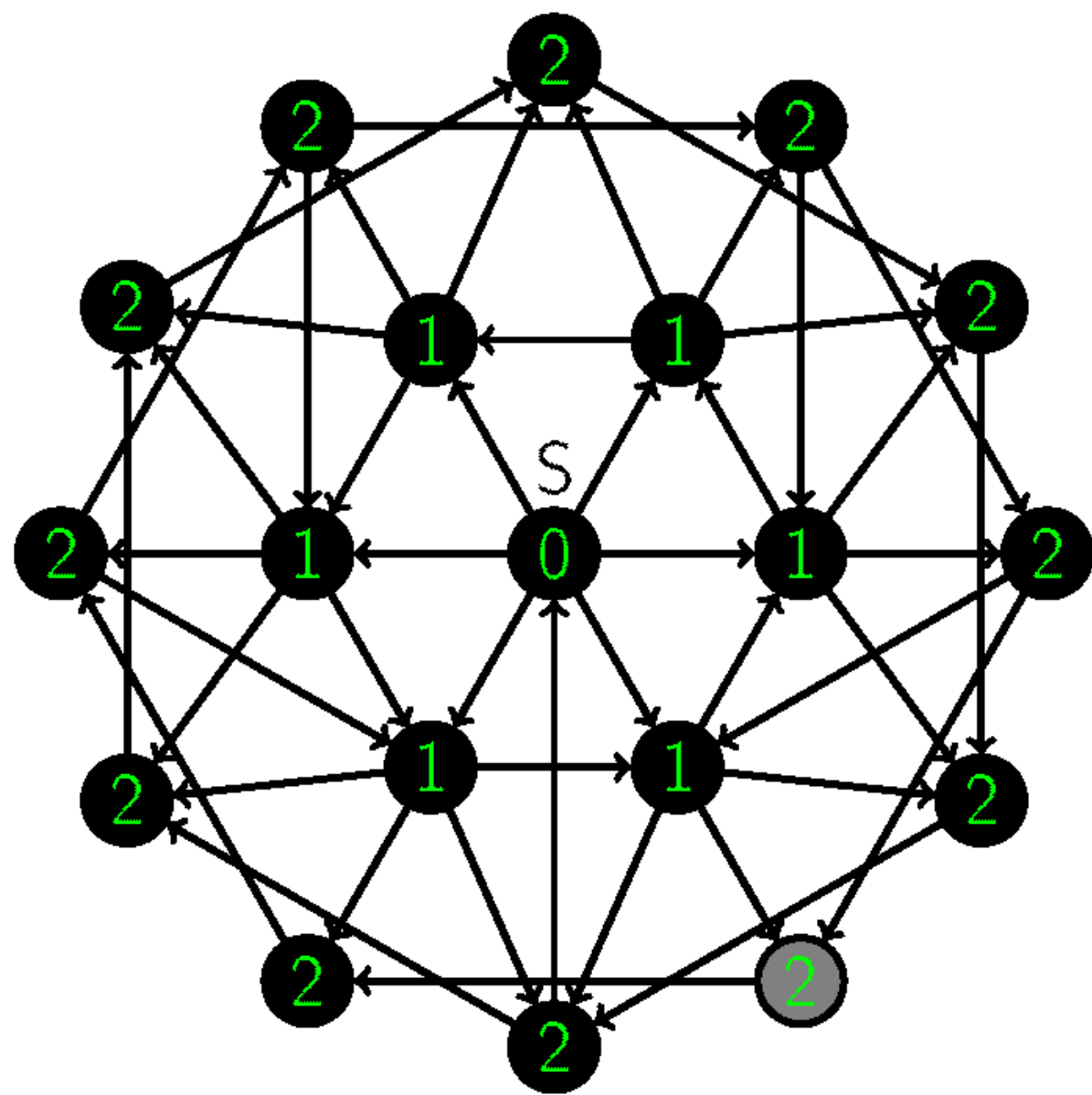


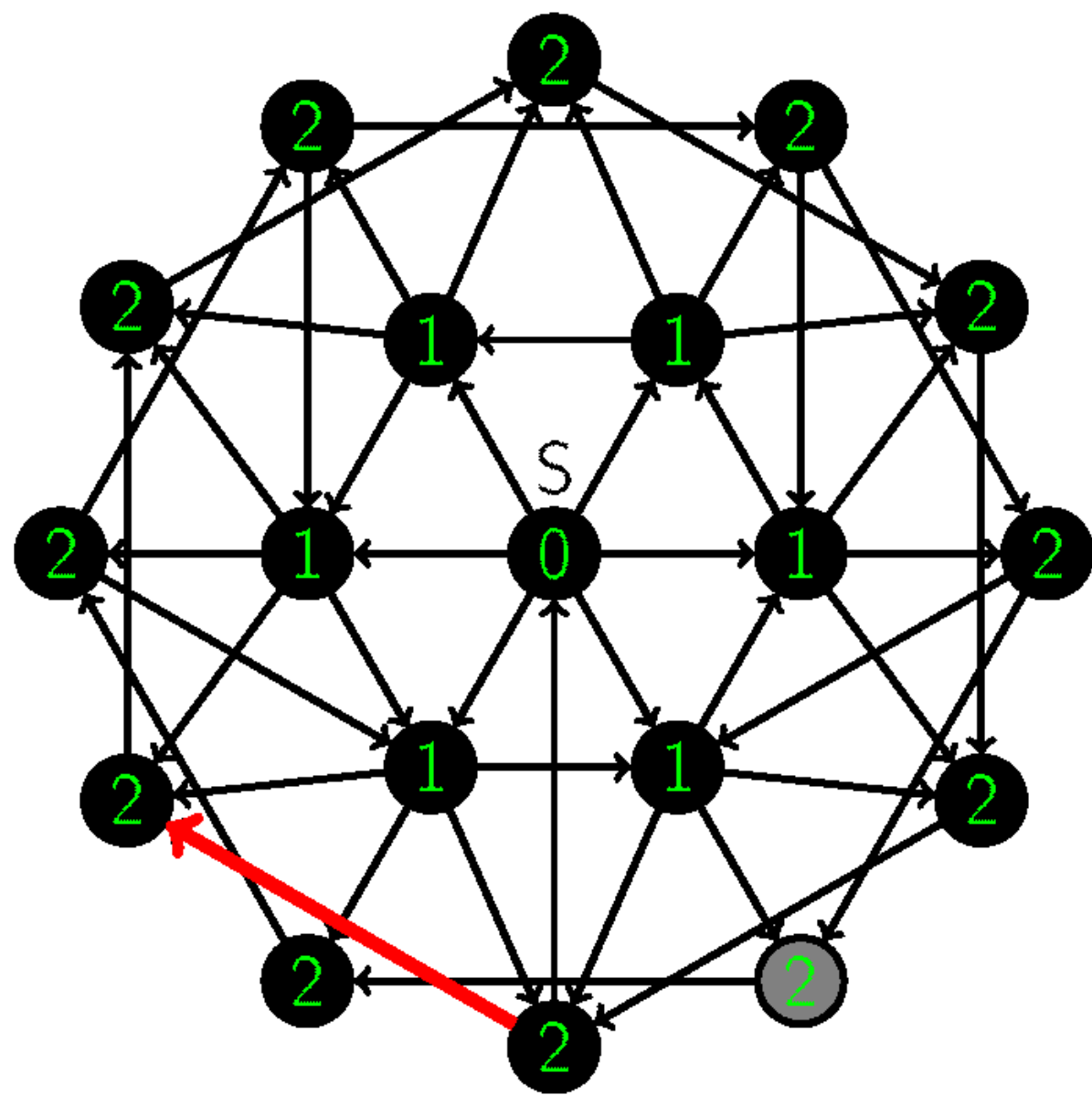


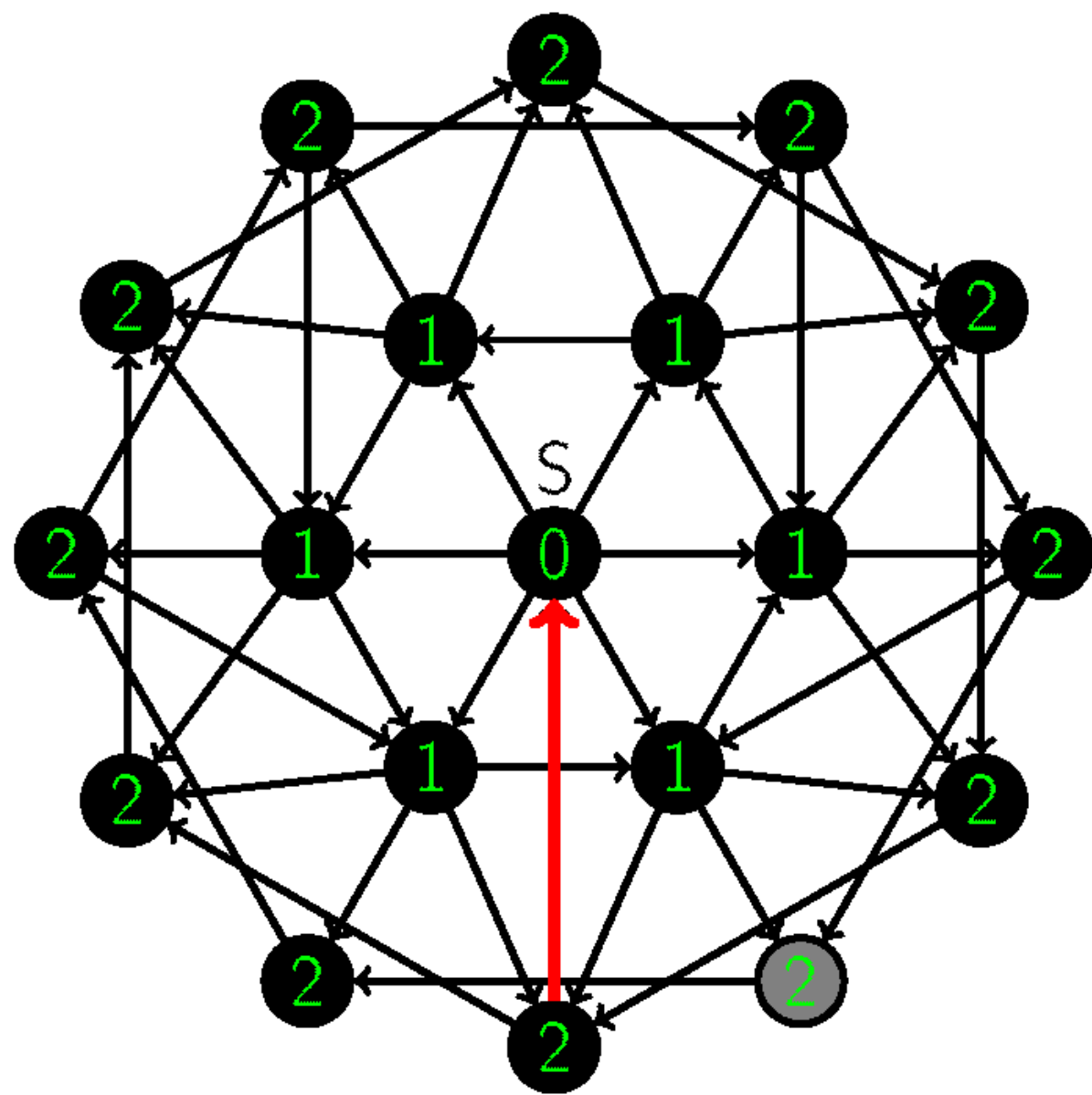


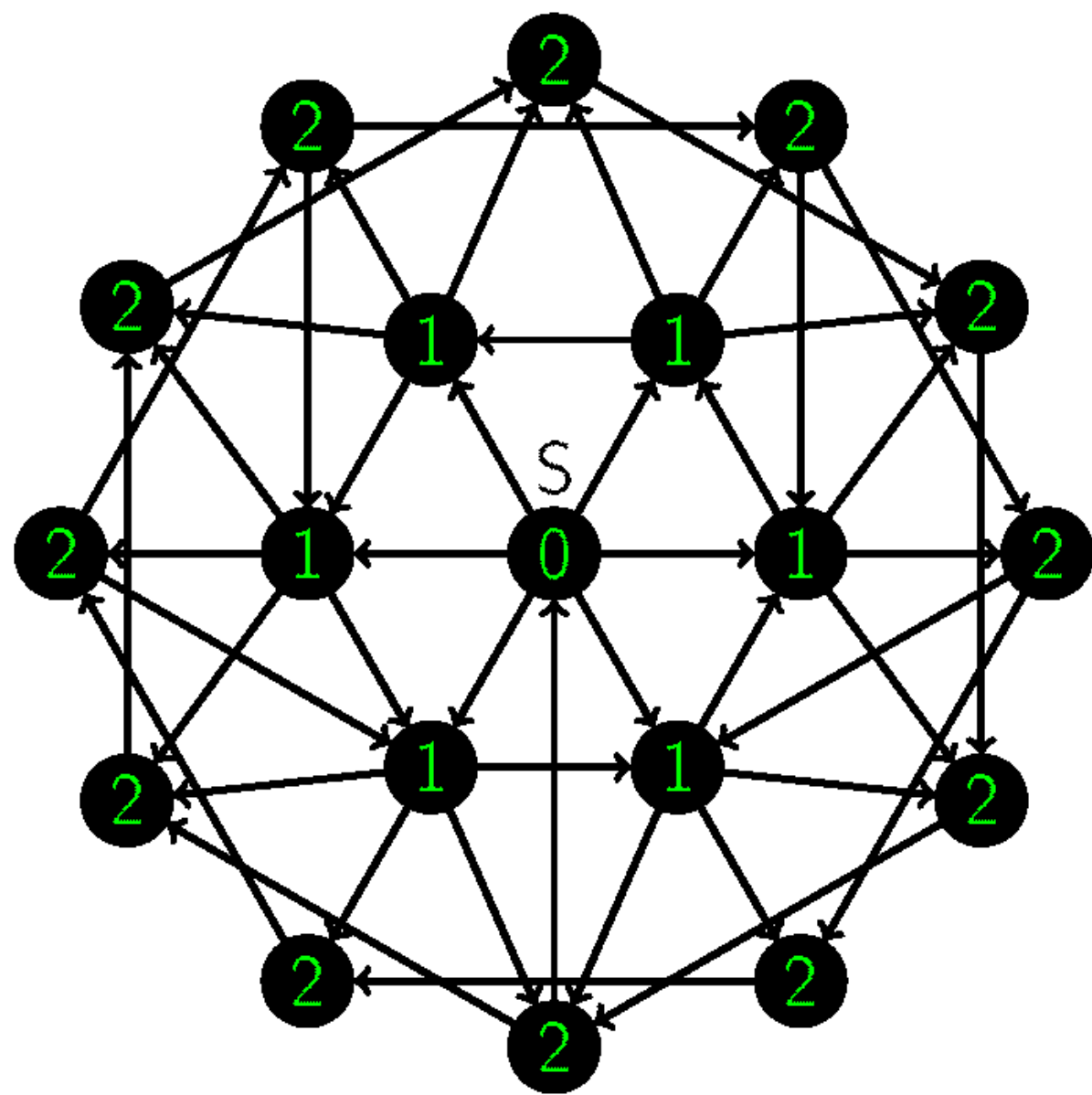


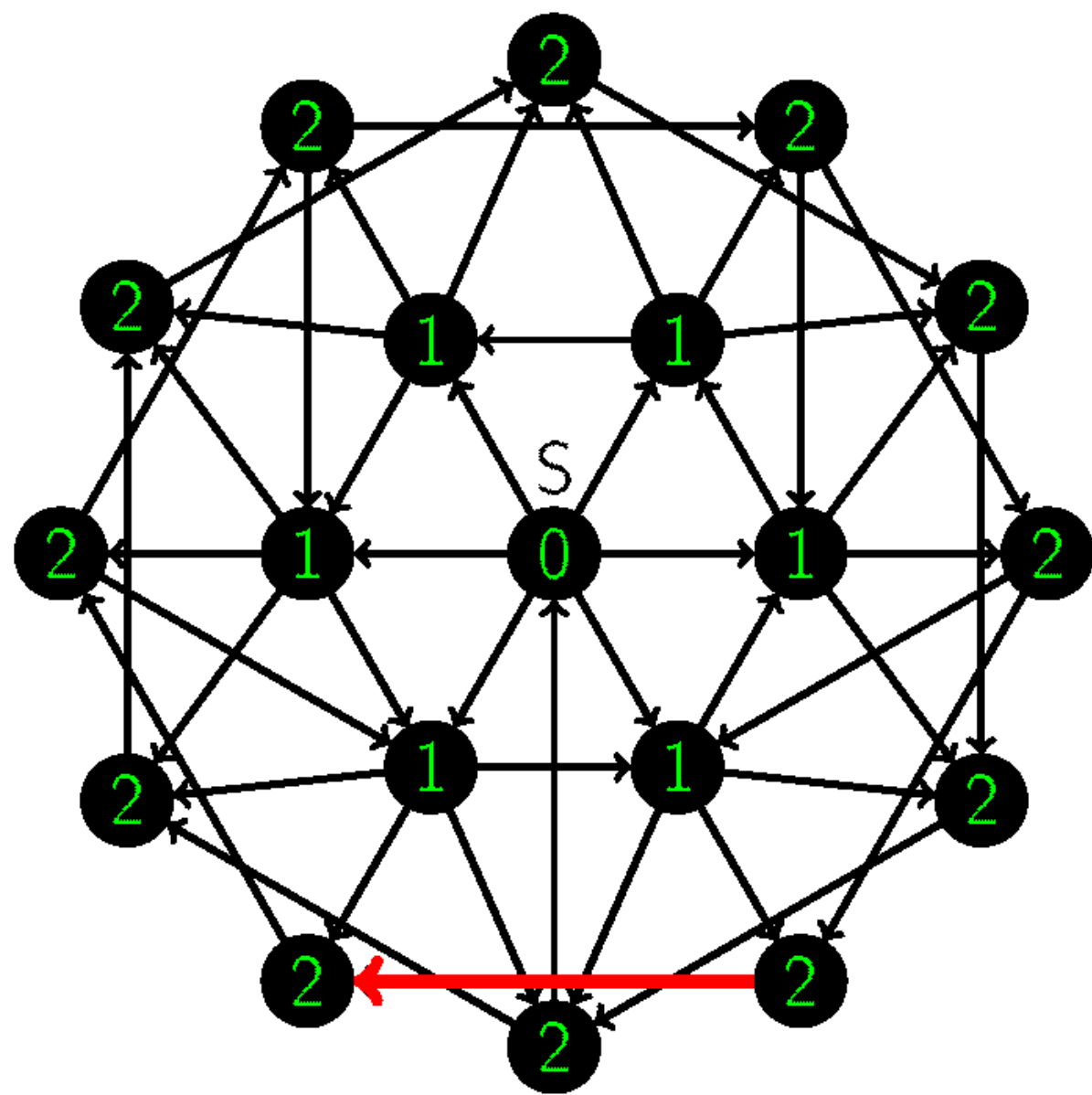


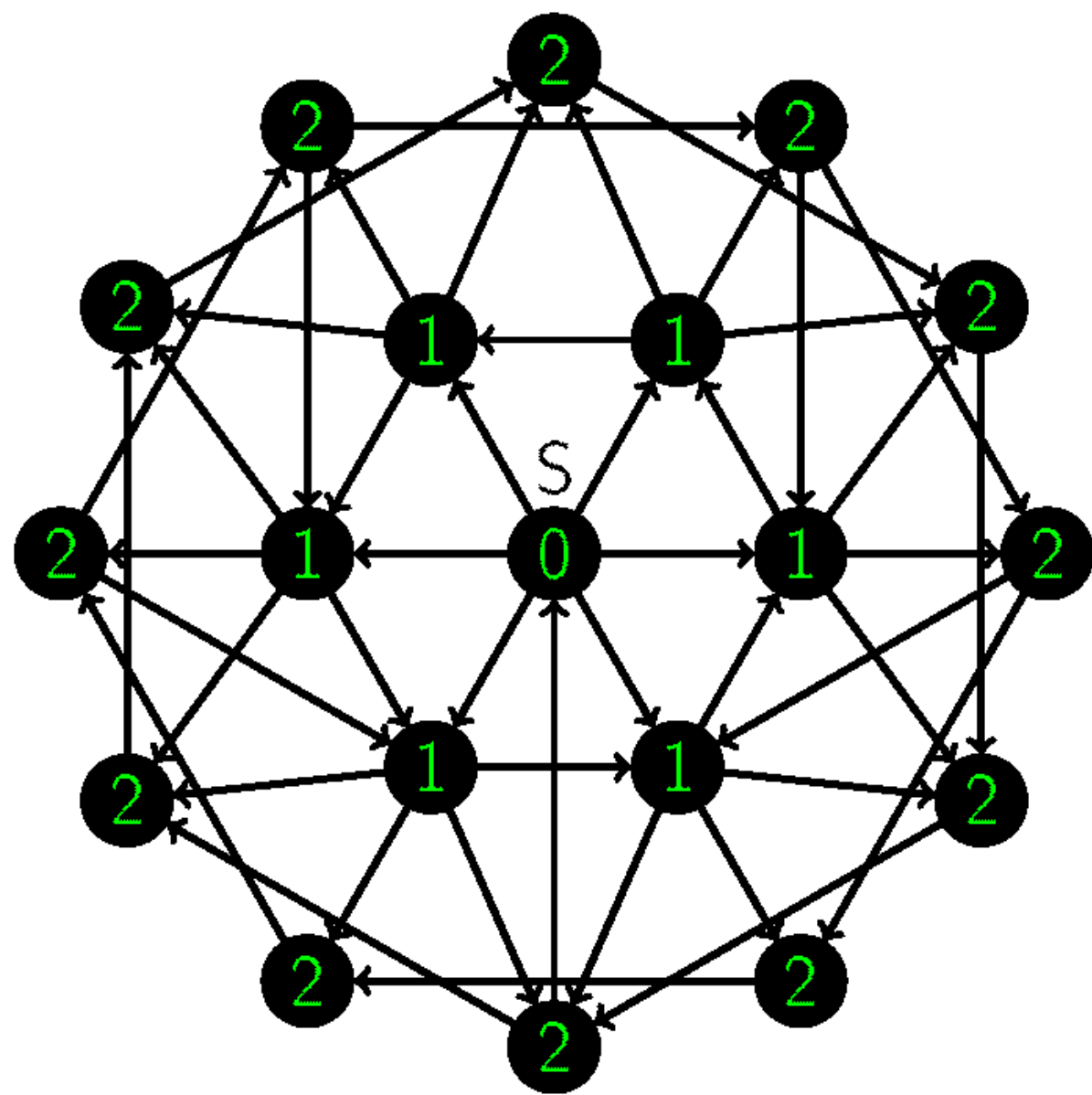










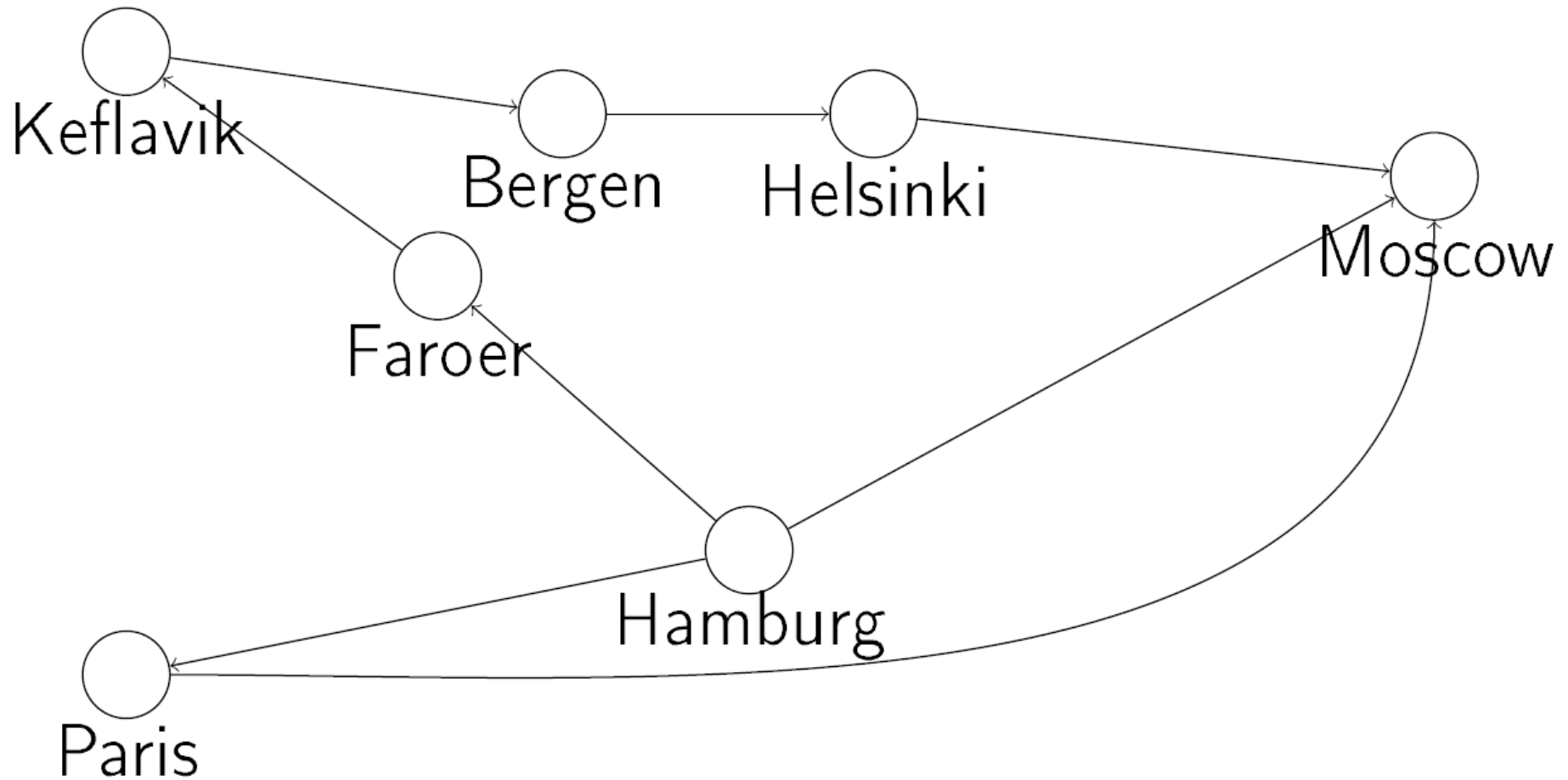


The most direct route

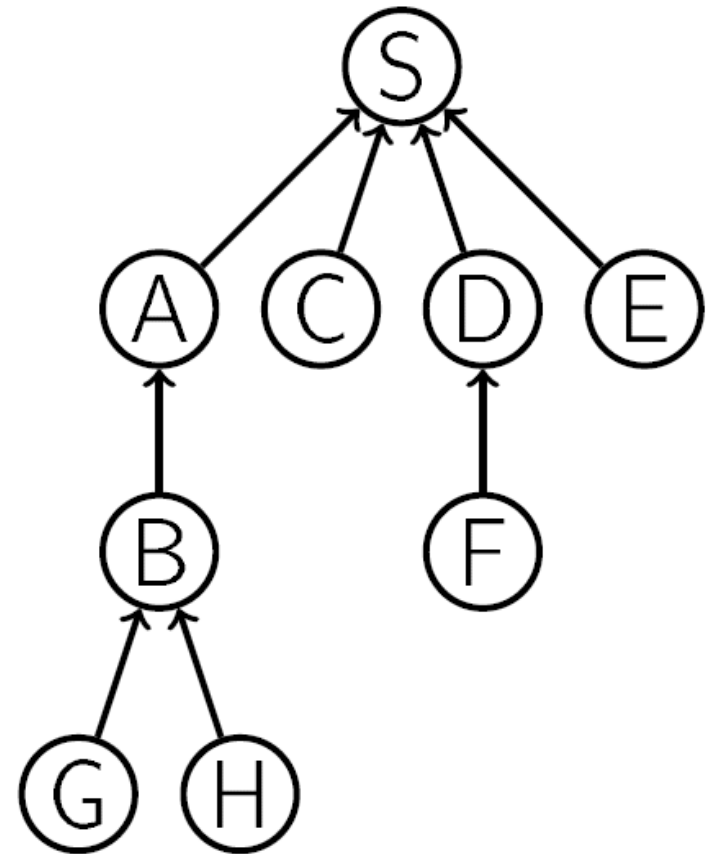
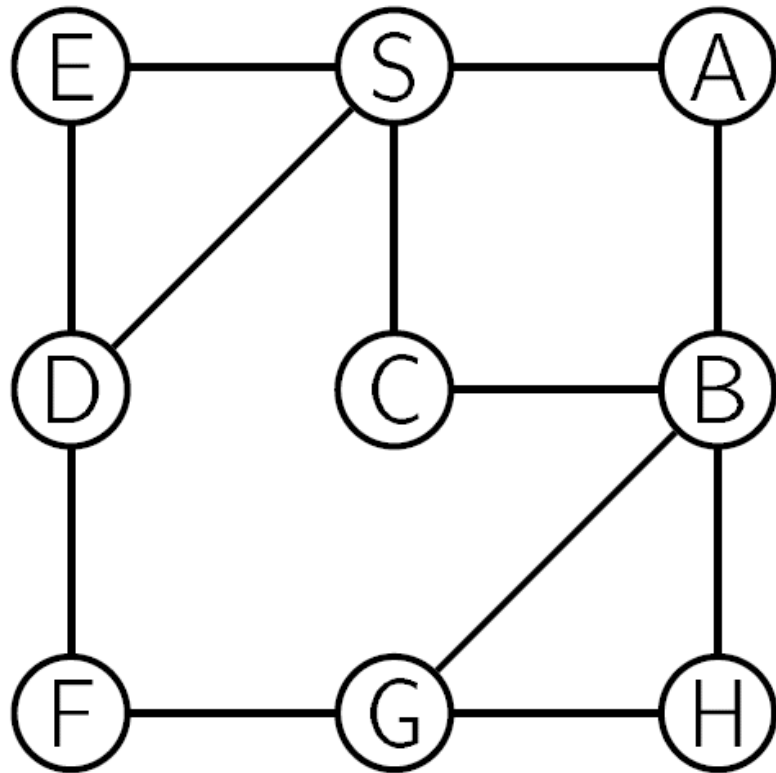
- What is the minimum number of flight segments to get from Hamburg to Moscow?



Which route has minimum flight segment?



Shortest-path tree



Constructing shortest-path tree

BFS(G, S)

```
for all  $u \in V$ :  
     $\text{dist}[u] \leftarrow \infty$ ,  $\text{prev}[u] \leftarrow \text{nil}$   
 $\text{dist}[S] \leftarrow 0$   
 $Q \leftarrow \{S\}$  {queue containing just  $S$ }  
while  $Q$  is not empty:  
     $u \leftarrow \text{Dequeue}(Q)$   
    for all  $(u, v) \in E$ :  
        if  $\text{dist}[v] = \infty$ :  
             $\text{Enqueue}(Q, v)$   
             $\text{dist}[v] \leftarrow \text{dist}[u] + 1$ ,  $\text{prev}[v] \leftarrow u$ 
```

Reconstructing Shortest Path

ReconstructPath(S, u, prev)

```
result  $\leftarrow$  empty
while  $u \neq S$ :
    result.append( $u$ )
     $u \leftarrow \text{prev}[u]$ 
return Reverse(result)
```

Conclusion

- Can find the minimum number of flight segments to get from one city to another
- Can reconstruct the optimal path
- Can build the tree of shortest paths from one origin
- Works in $O(|E| + |V|)$