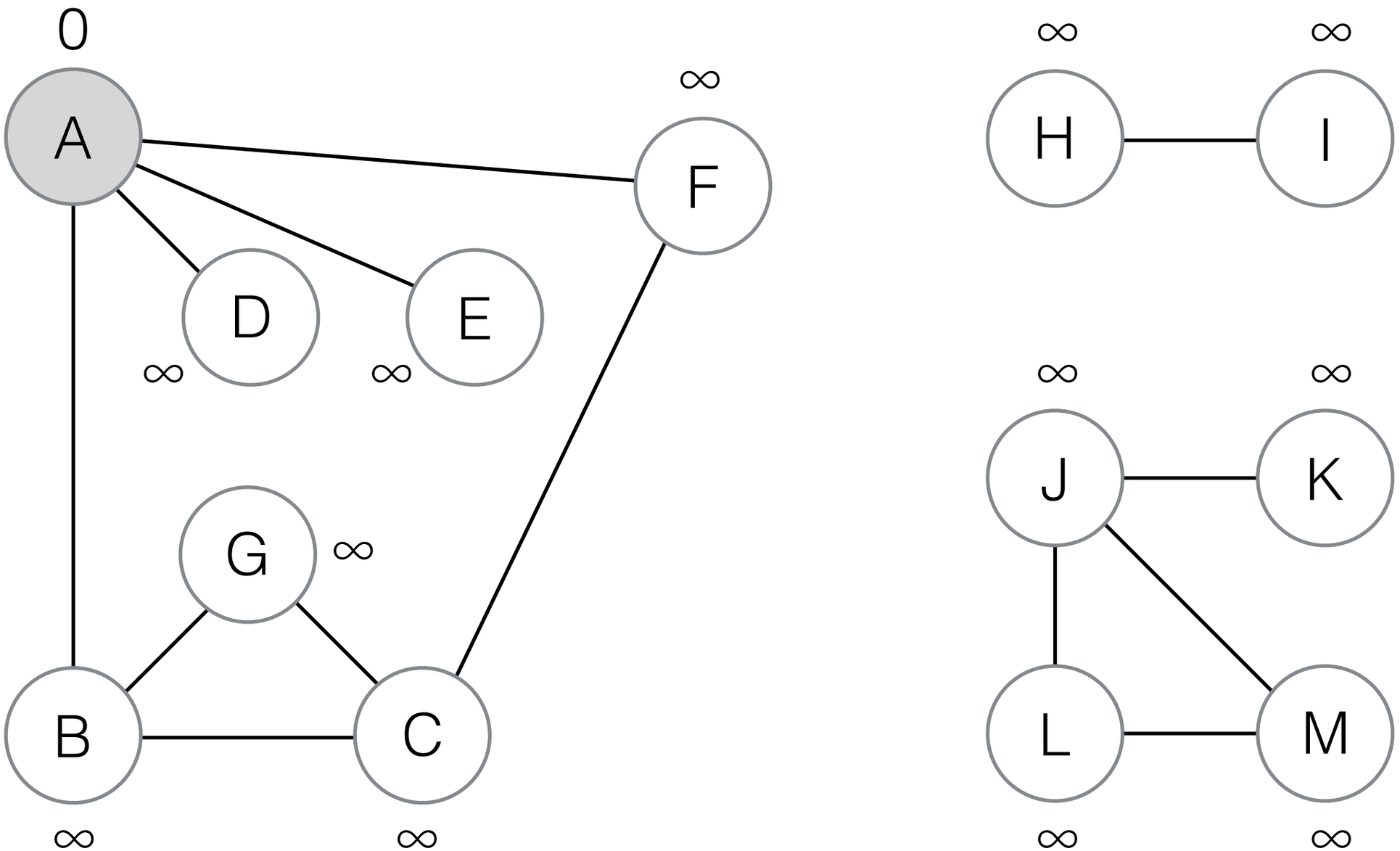
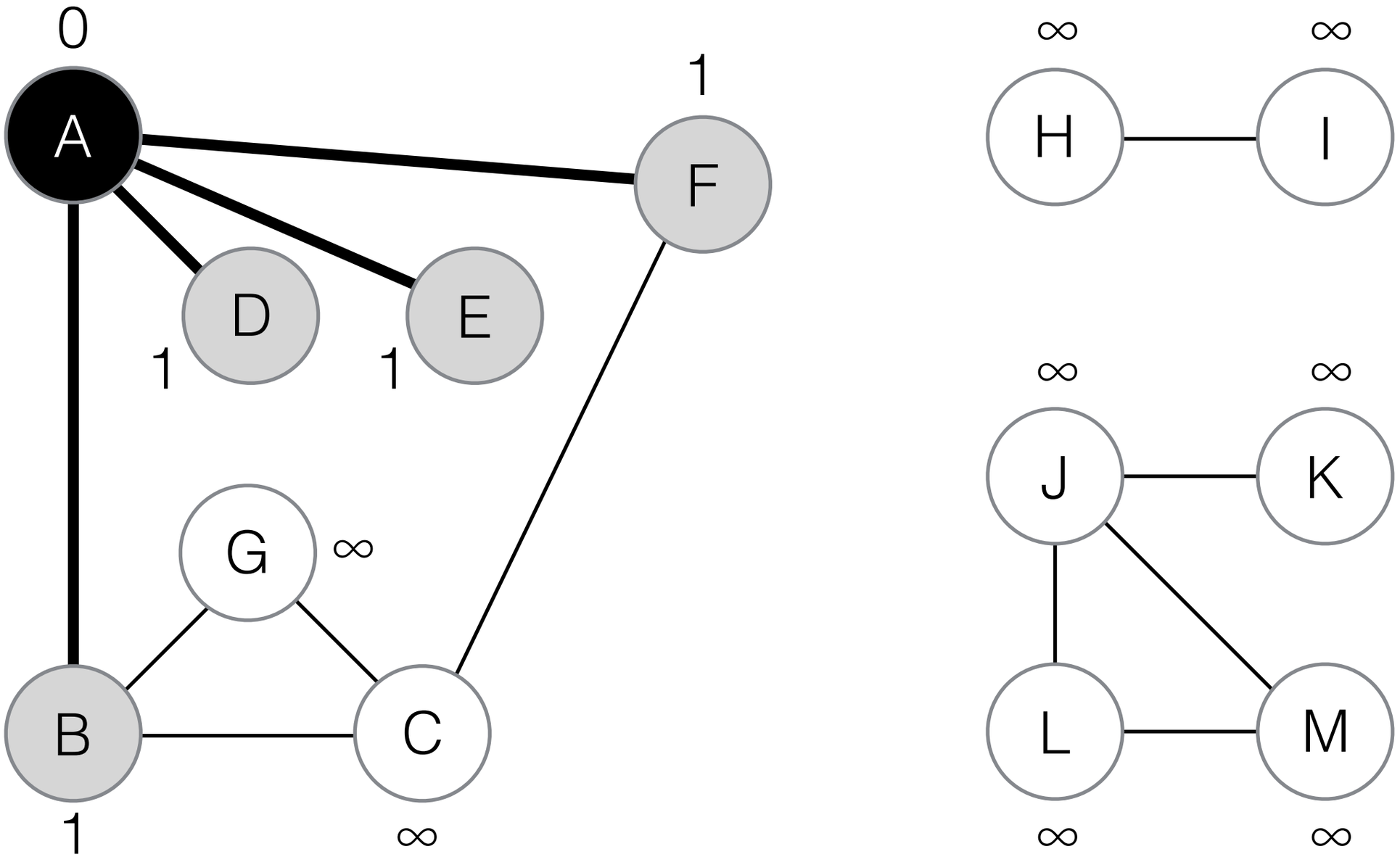
Breadth-First Search

```
BFS(G,s):

   let Q be an empty queue
   foreach node v in G except s
   |  color(v) = white
   |  dist(v) = ∞
   |  pred(v) = null
   color(s) = grey
   dist(s) = 0
   pred(s) = null
   Q.enqueue(s)

   while !Q.isEmpty()
   |  let u = Q.front()
   |  foreach neighbor v of u
   |  |  if color(v) == white
   |  |  |  color(v) = grey
   |  |  |  dist(v) = dist(u) + 1
   |  |  |  pred(v) = u
   |  |  |  Q.enqueue(v)
   |  Q.dequeue()
   |  color(u) = black

   return dist
```

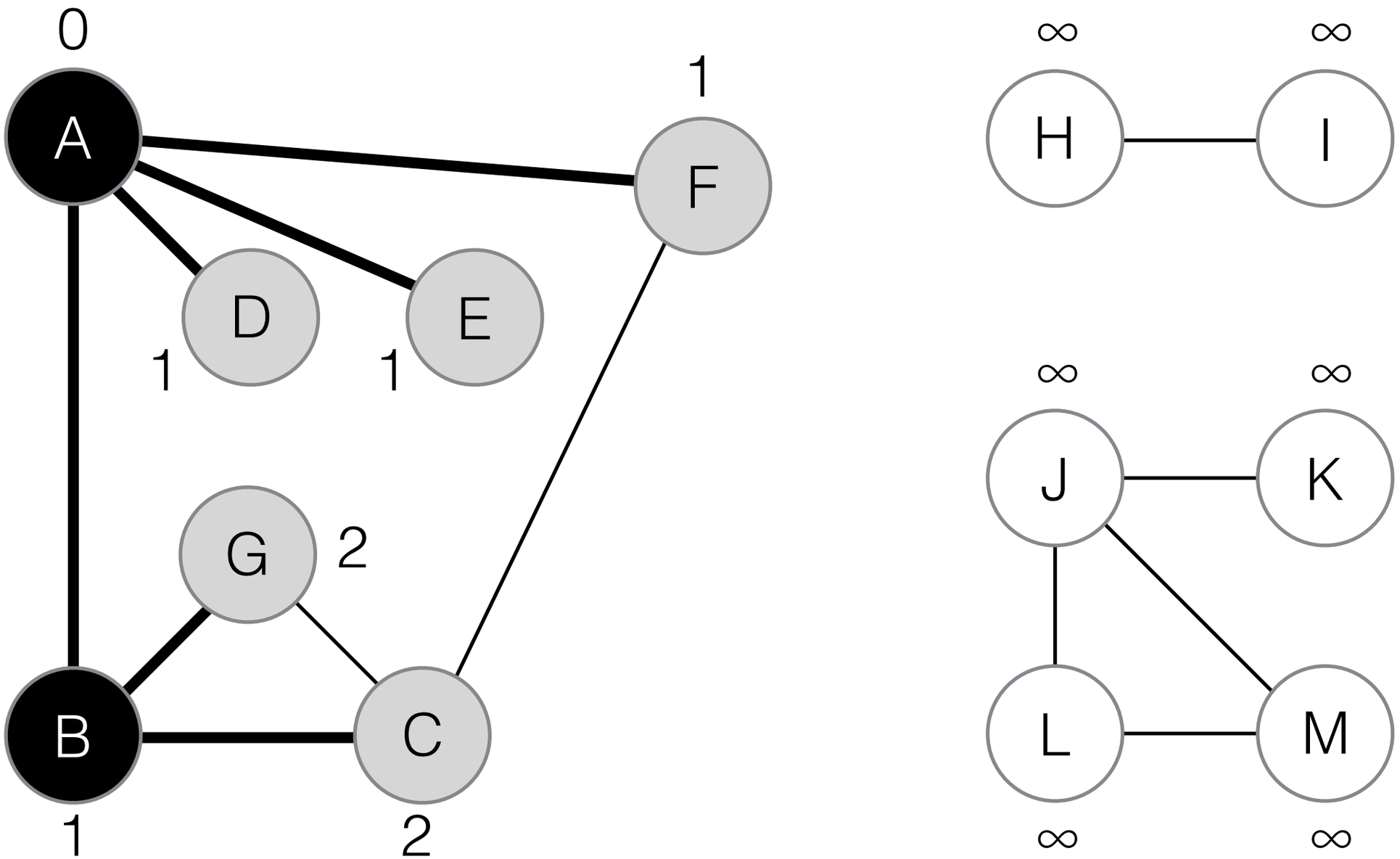
Q: A



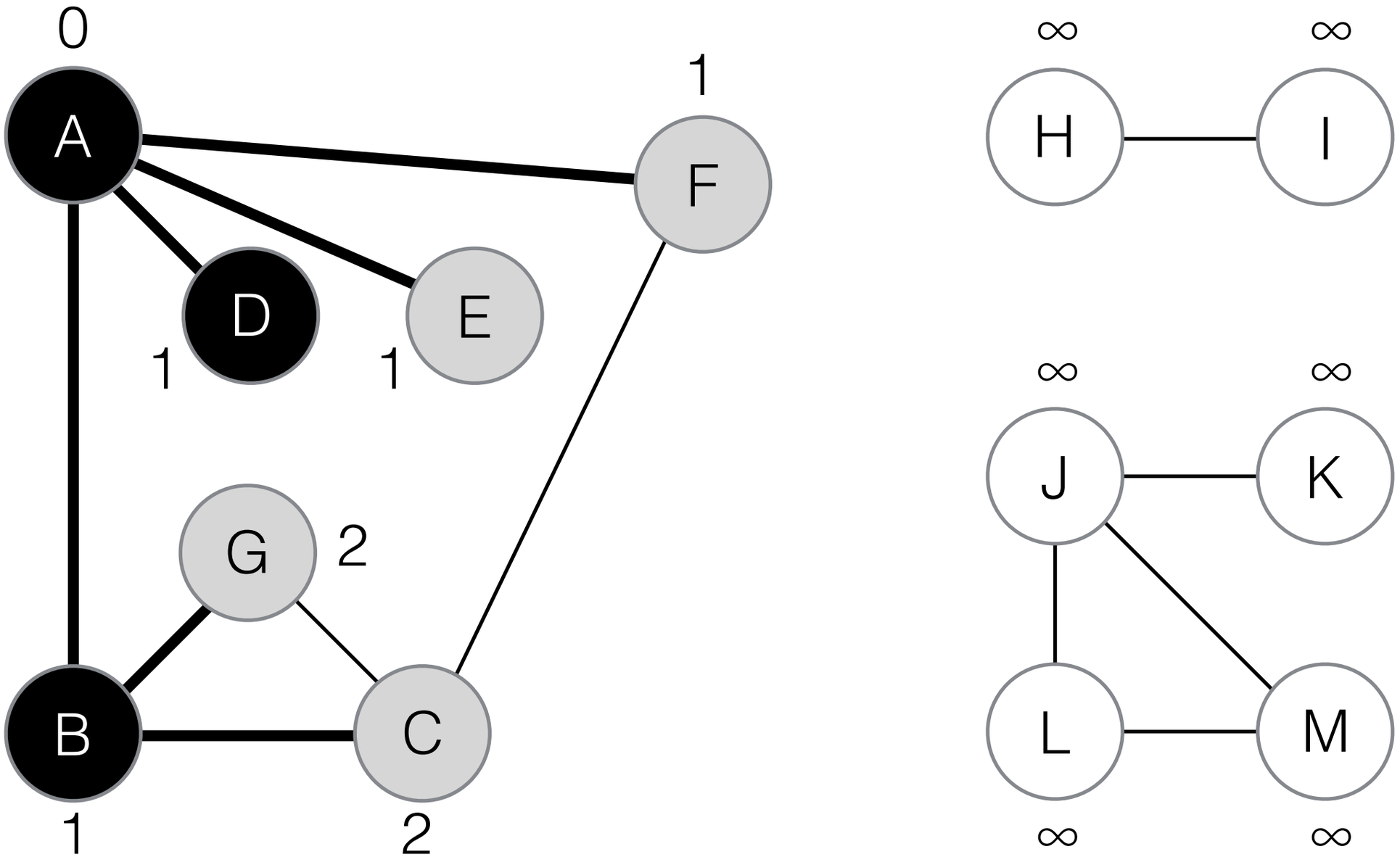
Q: | B | D | E | F |



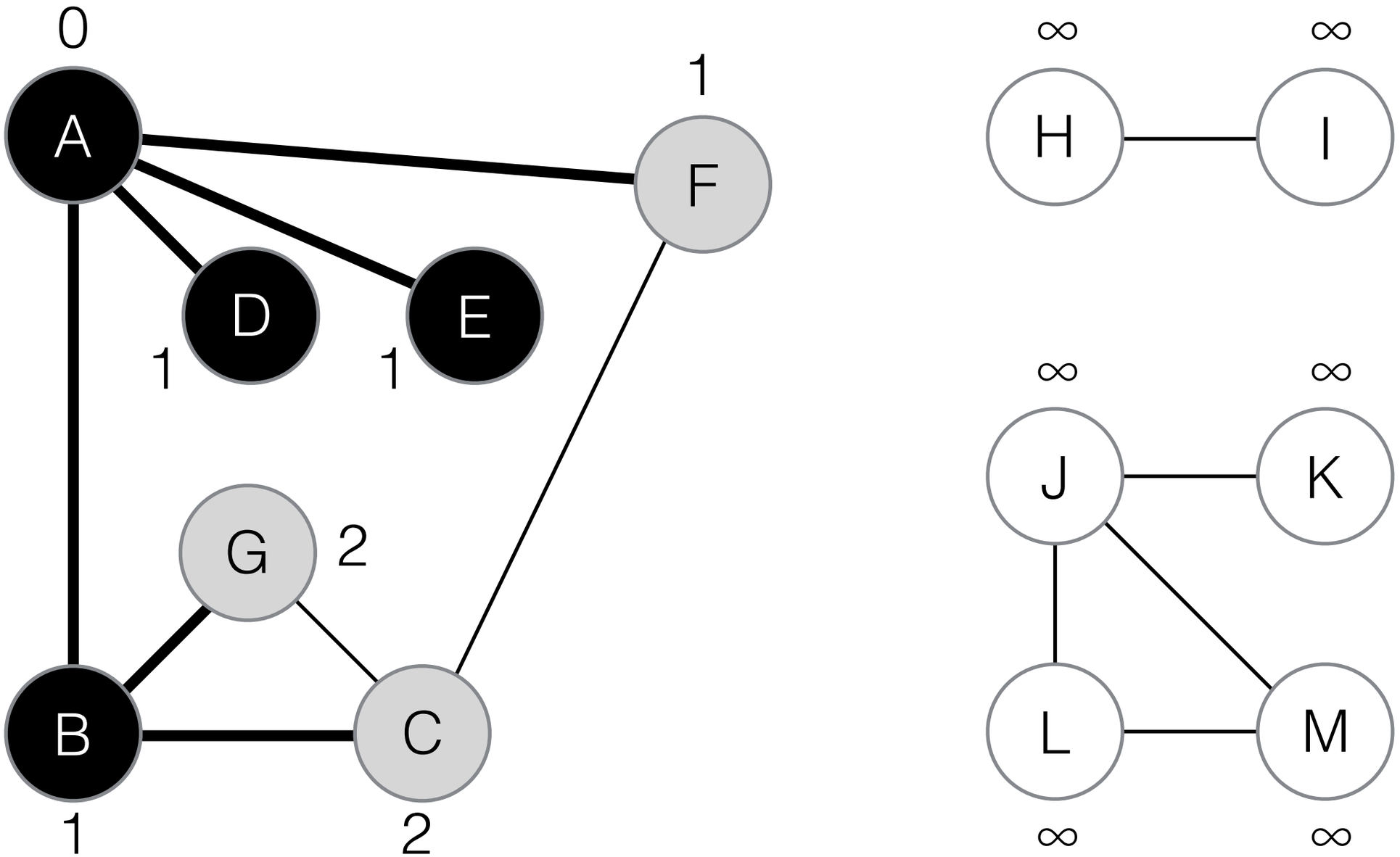
Q: | D | E | F | C | G |



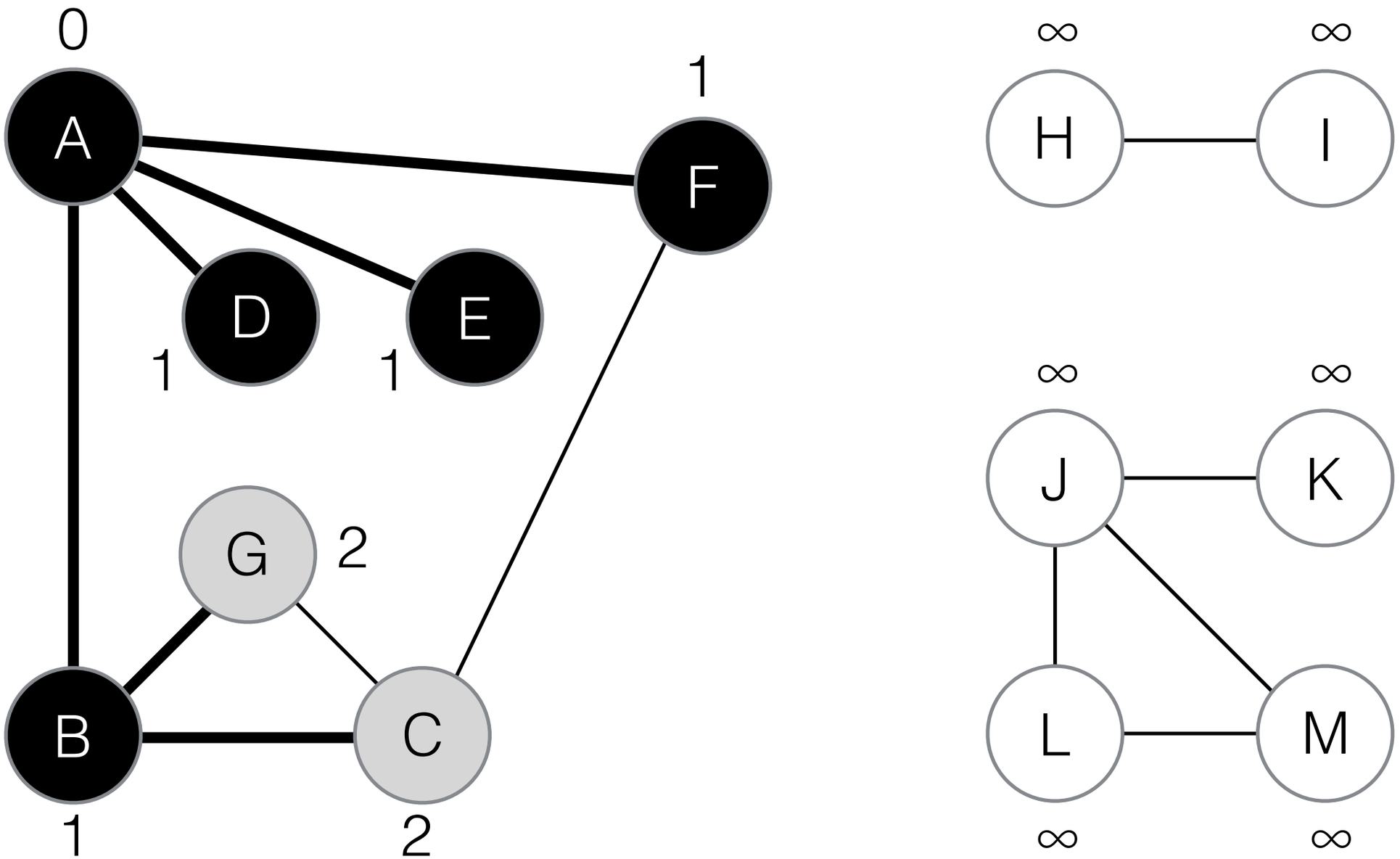
Q: | E | F | C | G |



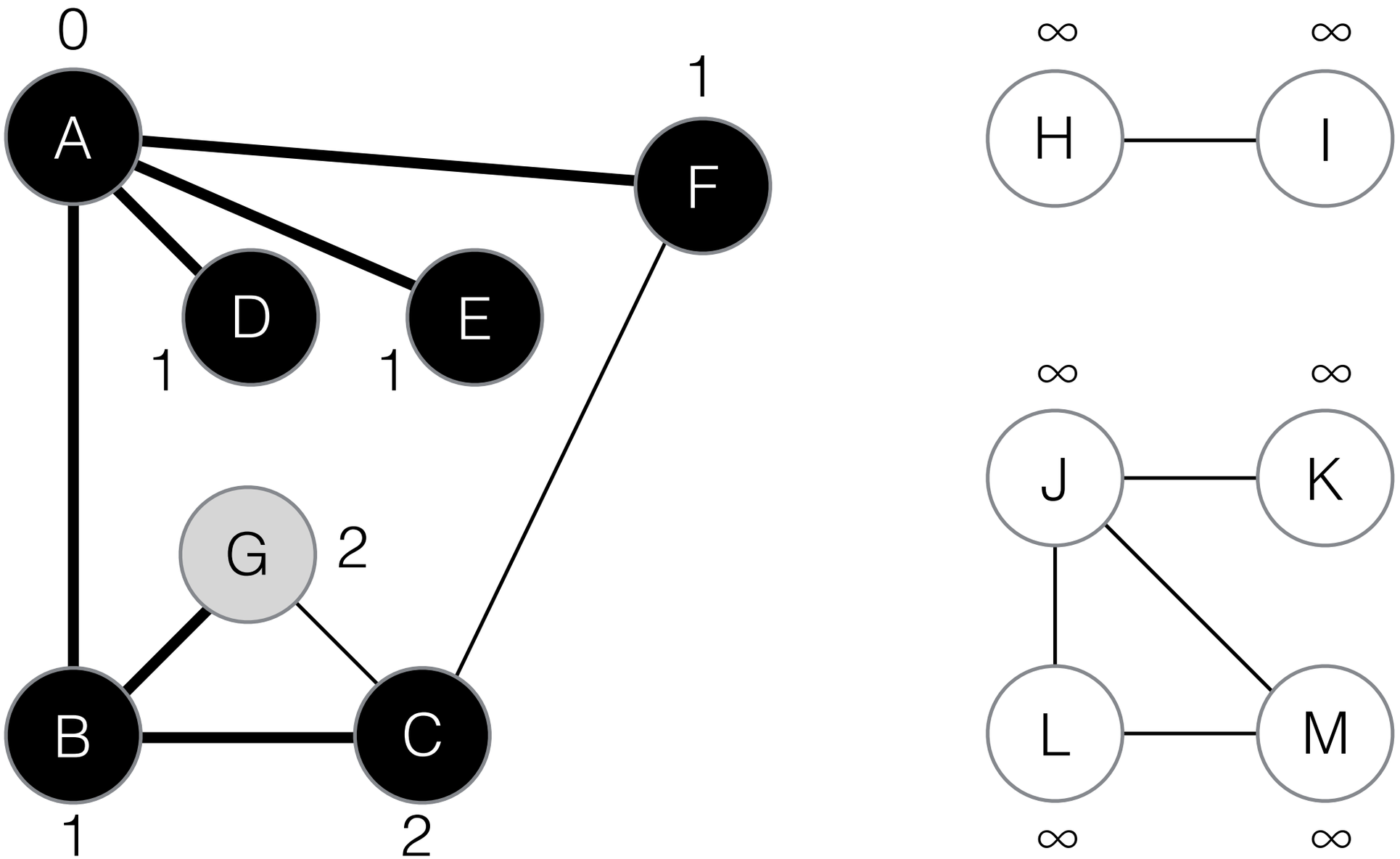
Q: | F | C | G |



Q: | C | G |



Q: G



Q: