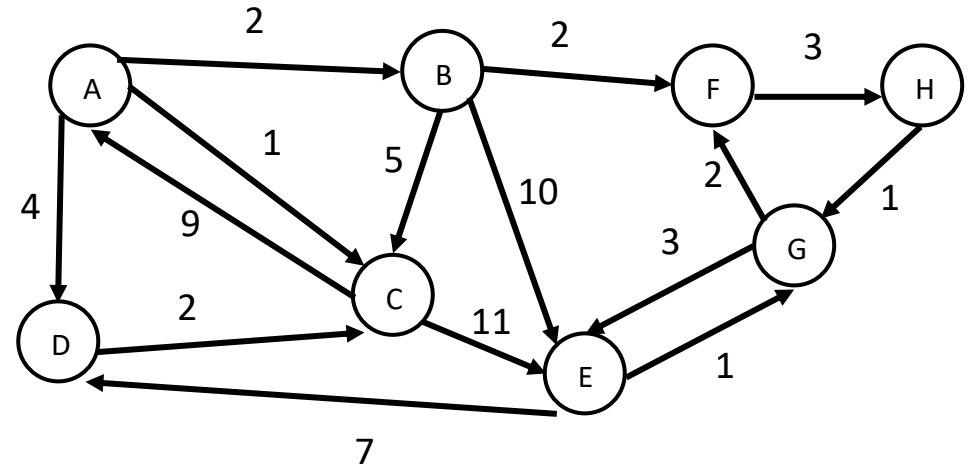


CSE 331 22sp Section 7 Handout

1. Recall the pseudocode for Dijkstra's algorithm, and consider the following graph below.

```

active = priority queue of paths.
finished = empty set of nodes.
add a path from start to itself to active
while active is non-empty:
    minPath = active.removeMin()
    minDest = destination node in minPath
    if minDest is dest:
        return minPath
    if minDest is in finished:
        continue
    for each edge e = (minDest, child):
        if child is not in finished:
            newPath = minPath + e
            add newPath to active
    add minDest to finished
    
```



Find the shortest path starting from **A** going to **E**. Record each update (push, pop) to the queue or any returns (true, false) in the table below.

| Node | Finished | Cost | Previous |
|------|----------|------|----------|
| A | | | |
| B | | | |
| C | | | |
| D | | | |
| E | | | |
| F | | | |
| G | | | |
| H | | | |