Part 2 – Theory

2.1

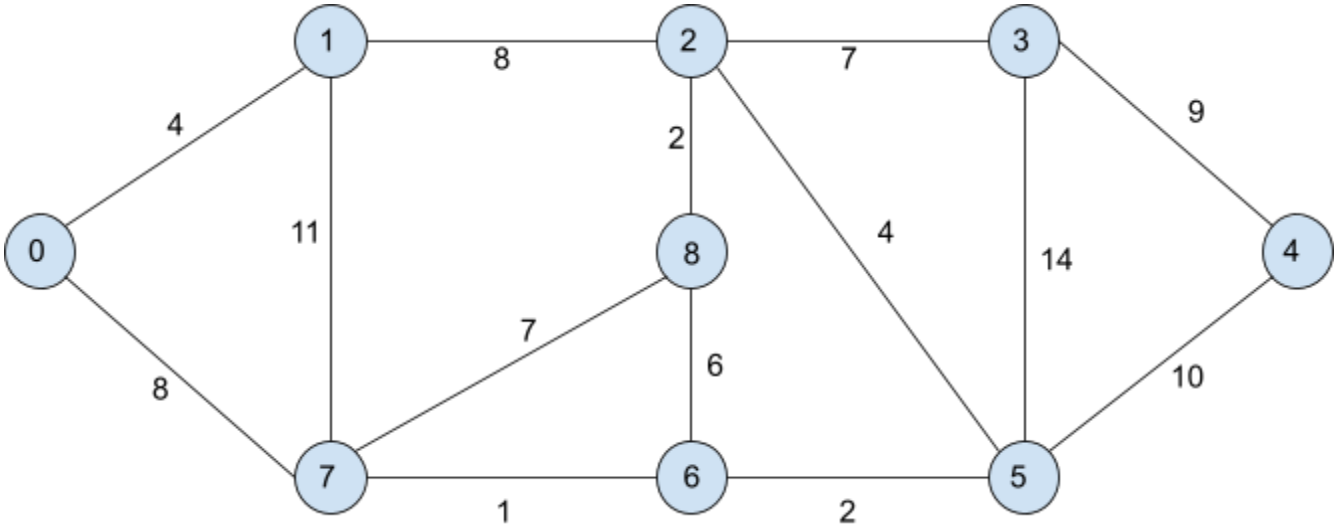
Based on the information provided, that is needed for every vertex, the following algorithm can be run for every vertex in G. The backing structure of this PQ is a minHeap.

```
Algorithm ModifiedShortestPath(G, v):
    initialize D[v] ← 0 and D[u] ← ∞ for each vertex v ≠ u
    let Q be a priority queue that contains all of the vertices of G using the D labels as keys.
    while Q ≠ ∅ do
        u ← Q.removeMinElement()
        If D[u] <= 30 then printCustomers(v)
        else break loop;
        for each vertex z adjacent to u such that z is in Q do
            if D[u] + w((u, z)) < D[z] then
                D[z] ←D[u] + w((u, z))
                change the key value of z in Q to D[z]
```

ModifiedShortestPath(G,v) is like Dijkstra's shortest path, but stops when it reaches the first vertex at a distance larger than 30. The analysis is like that one, worst case O(m*log(n)); or more precisely, if n1 is the number of vertices at distance <= 30 from v, and m1 is the total number of edges such that one of its ends is at distance <= 30 from v, then this runs in O(n + m1*log(n1)). If running once per vertex, then the running time is O(n*m*log(n)).

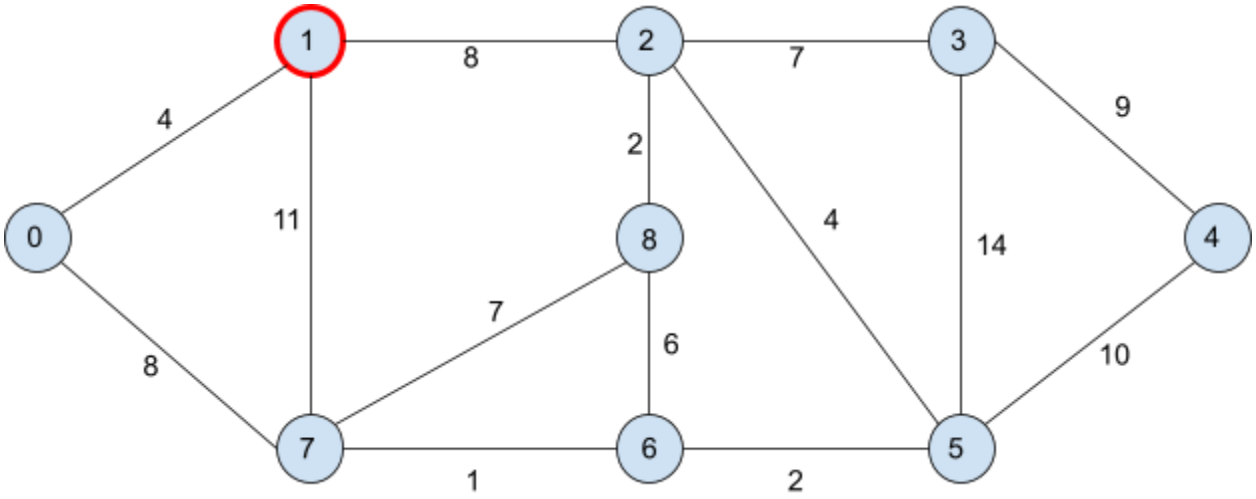
2.2

Original graph



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

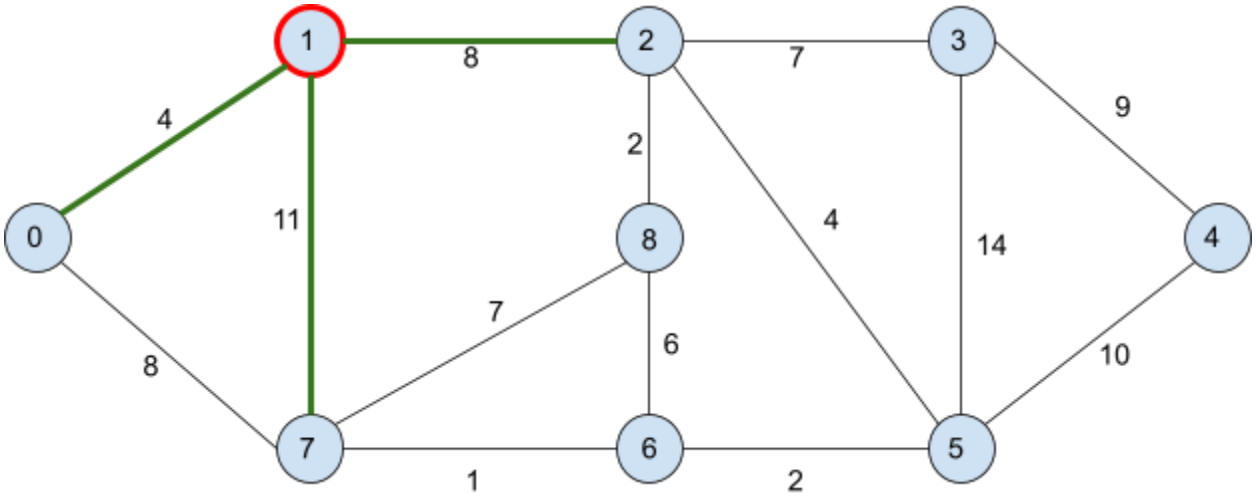
Current Node:
Unvisited Nodes: {0, 1, 2, 3, 4, 5, 6, 7, 8}
Visited Nodes (in order): {1}



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 1
Unvisited Nodes: {0, ~~1~~, 2, 3, 4, 5, 6, 7, 8}
Visited Nodes (in order): {1}

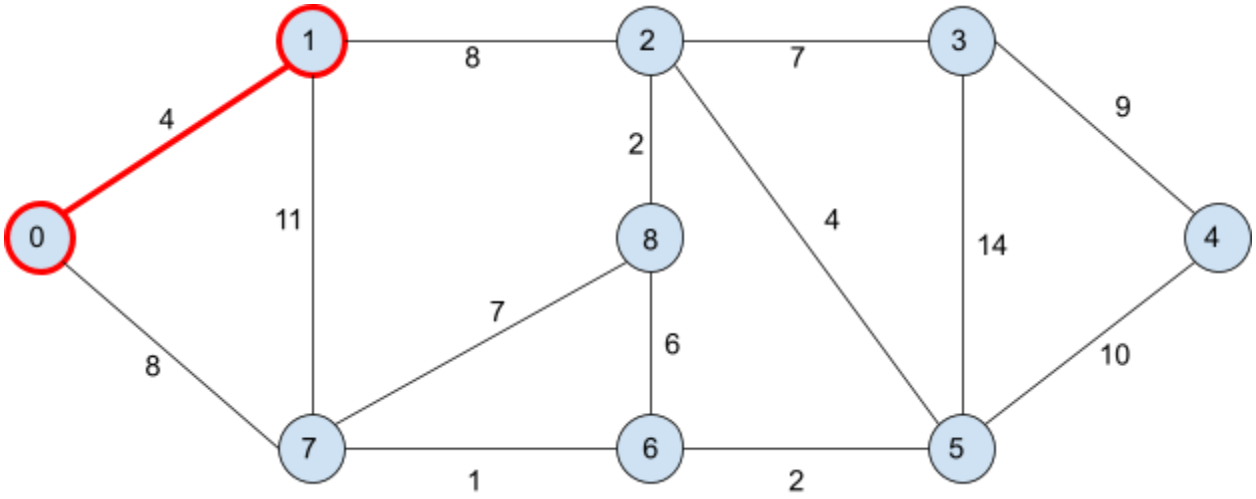
Red vertex => discovered vertex (new, never visited edges that lead to a newly-discovered node)
Red edge => discovered edge
Black edge => undiscovered edge



Vertex (edge-adjacent vertex)
0: **(0-1)**, (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), **(2-1)**
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), **(7-1)**
8: (8-2), (8-6), (8-7)

Current Node: 1
Unvisited Nodes: {0, ~~1~~, 2, 3, 4, 5, 6, 7, 8}
Visited Nodes (in order): {1}

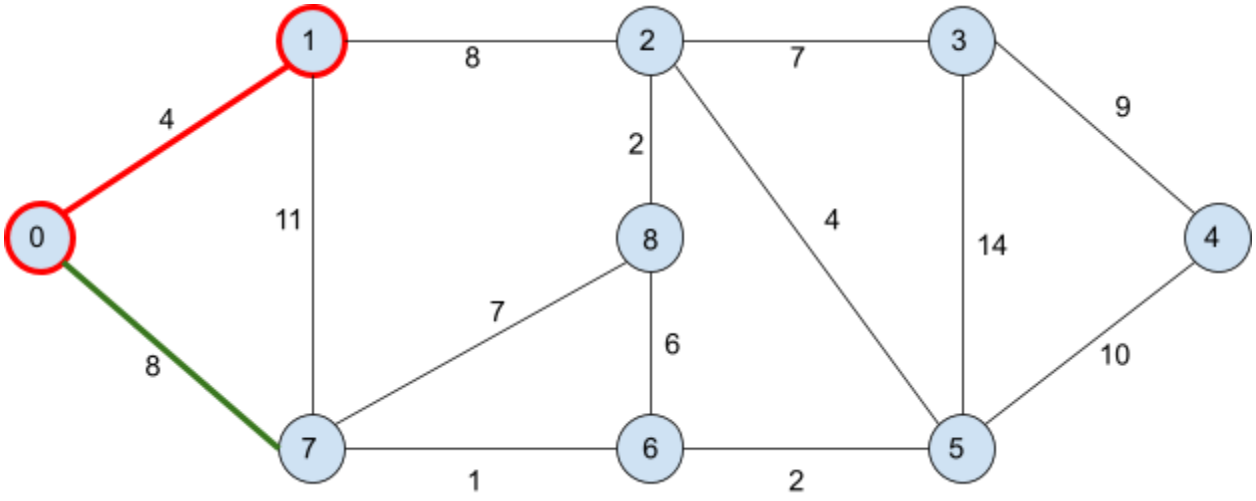
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 0
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, 6, 7, 8}
Visited Nodes (in order): {1, 0}

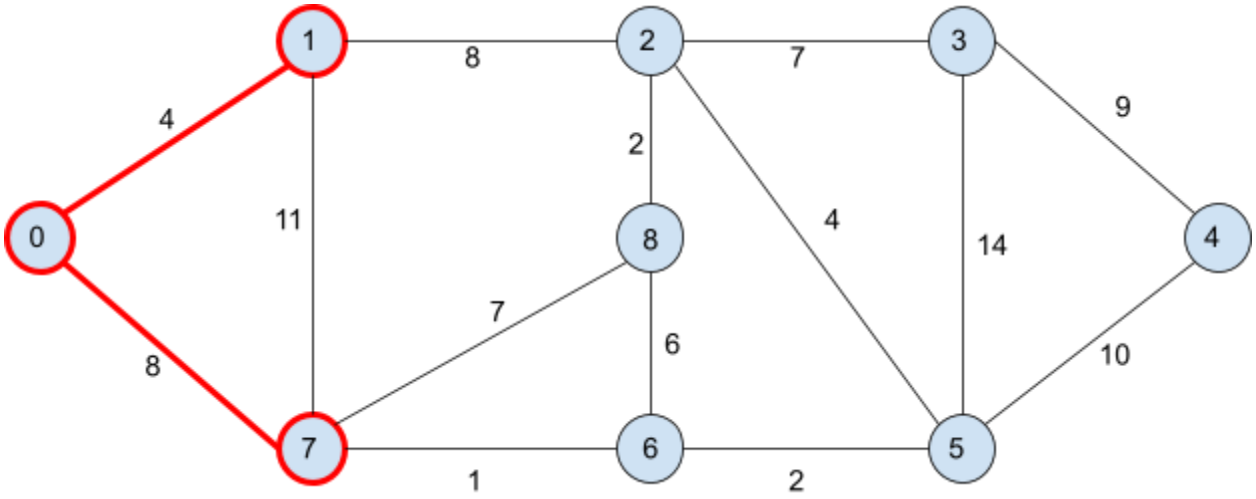
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), **(0-7)**
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 0
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, 6, 7, 8}
Visited Nodes (in order): {1, 0}

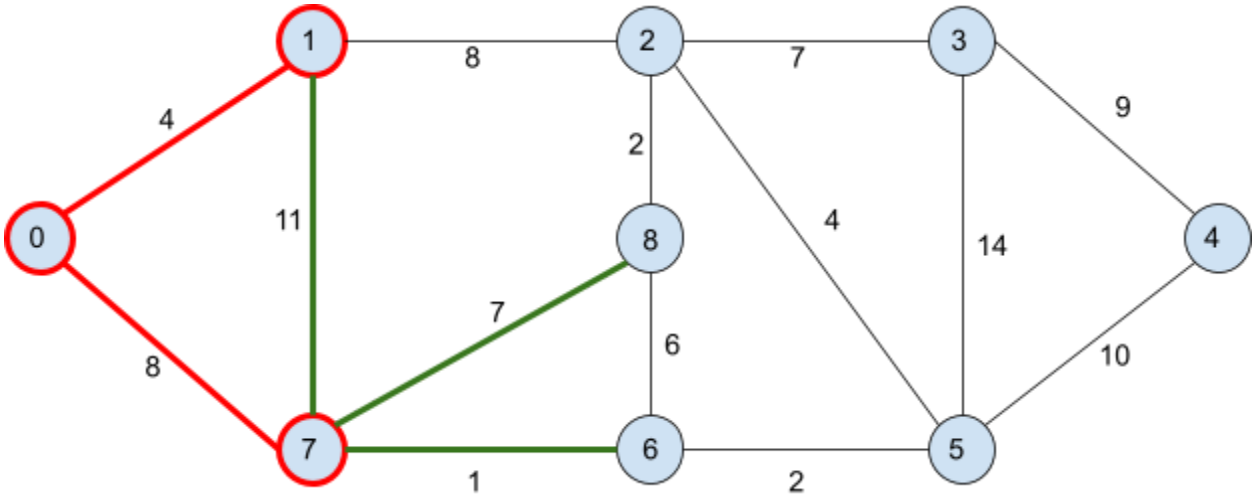
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 7
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, 6, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7}

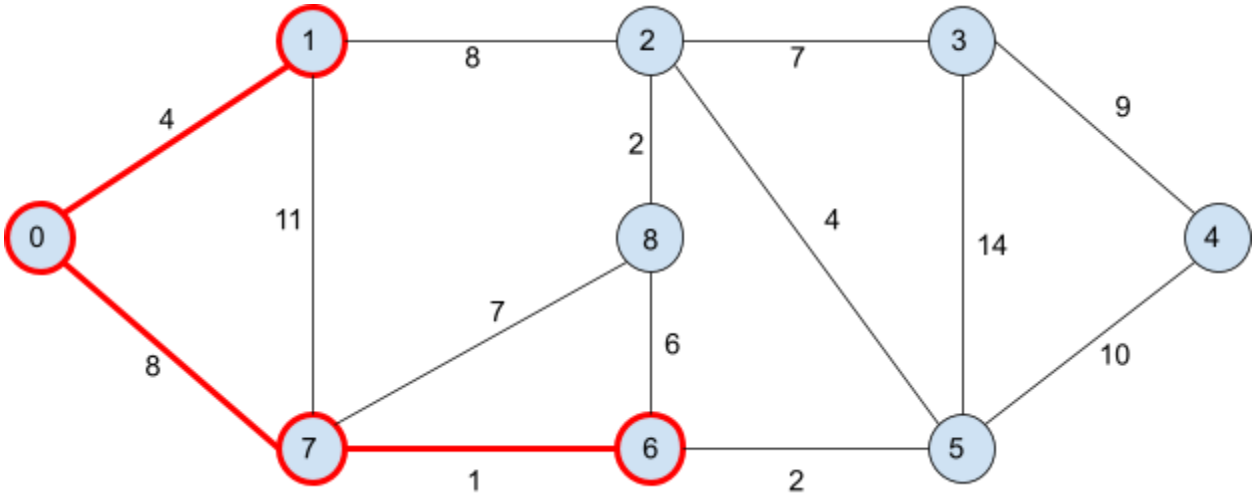
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), **(0-7)**
1: (1-0), (1-2), **(1-7)**
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: **(6-7)**, (6-5), (6-8)
7: (7-6), (7-8), **(7-0)**, **(7-1)**
8: (8-2), (8-6), **(8-7)**

Current Node: 7
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, 6, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7}

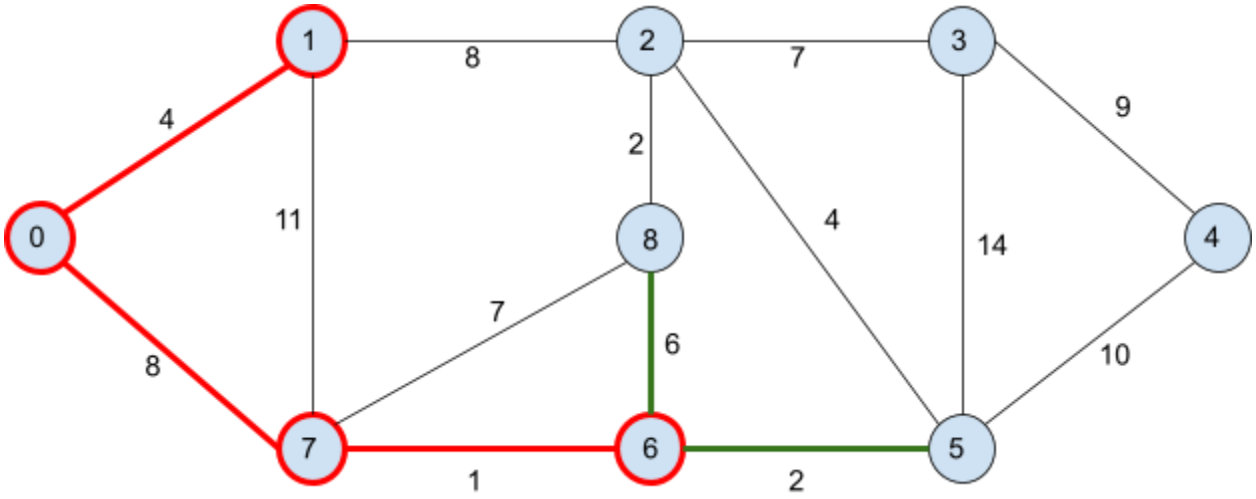
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), **(0-7)**
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 6
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6}

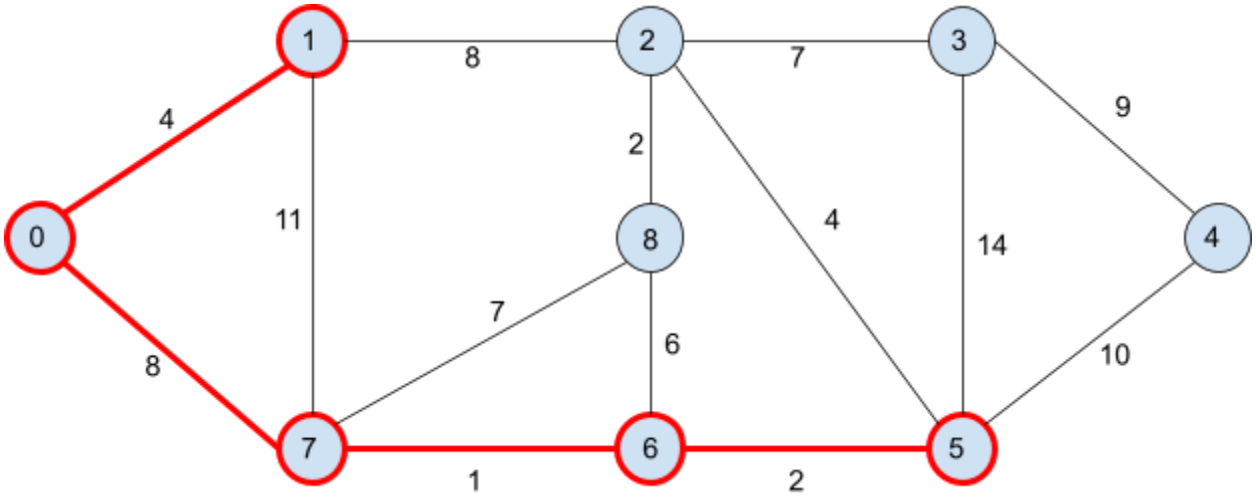
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: **(5-6)**, (5-2), (5-4), (5-3)
6: (6-7), **(6-5), (6-8)**
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), **(8-6)**, (8-7)

Current Node: 6
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, 5, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6}

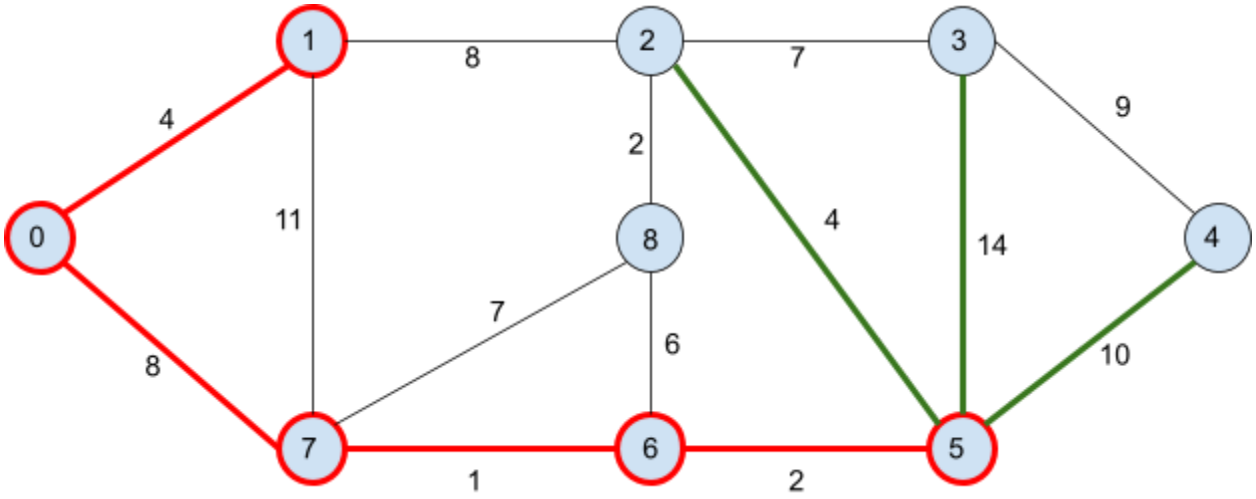
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5),** (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 5
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, ~~5~~, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6, 5}

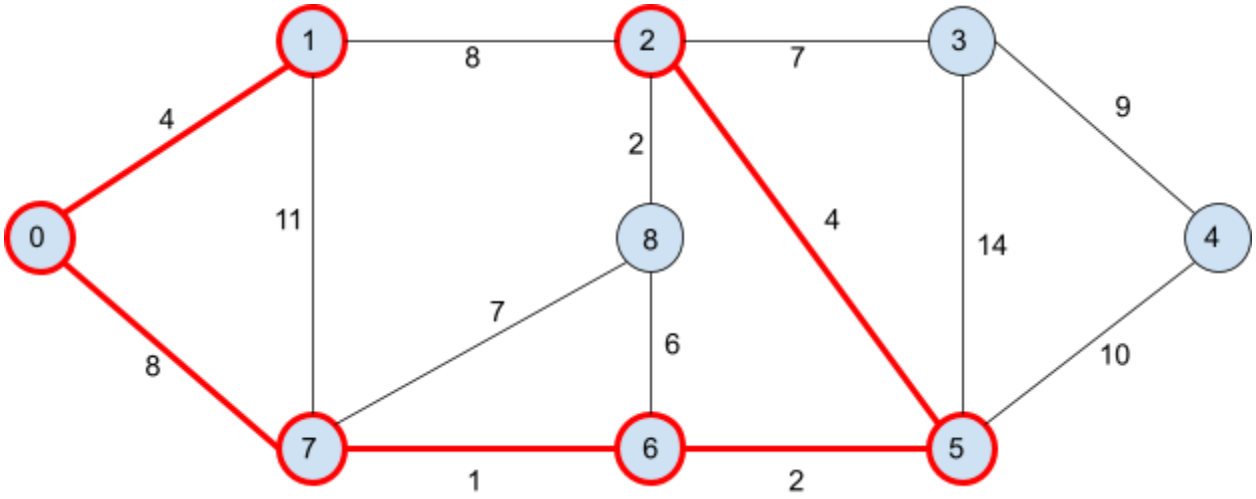
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), **(2-5)**, (2-3), (2-1)
3: (3-2), (3-4), **(3-5)**
4: (4-3), **(4-5)**
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5),** (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 5
Unvisited Nodes: {~~0~~, ~~1~~, 2, 3, 4, ~~5~~, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6, 5}

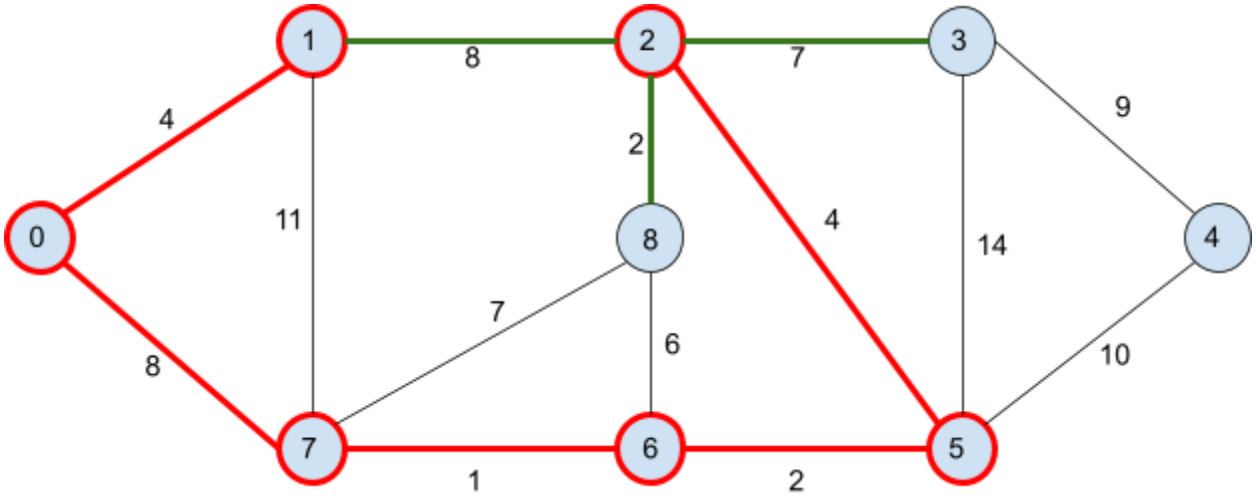
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), **(2-5)**, (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5),** (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 2
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2}

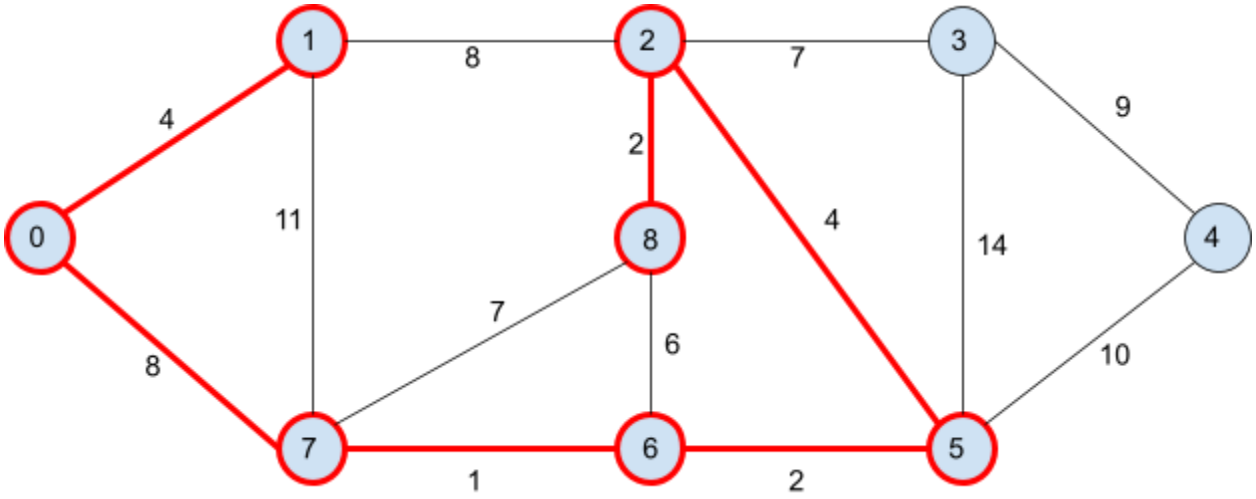
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), **(1-2)**, (1-7)
2: (2-8), **(2-5)**, **(2-3), (2-1)**
3: **(3-2)**, (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5),** (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: **(8-2)**, (8-6), (8-7)

Current Node: 2
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, 8}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2}

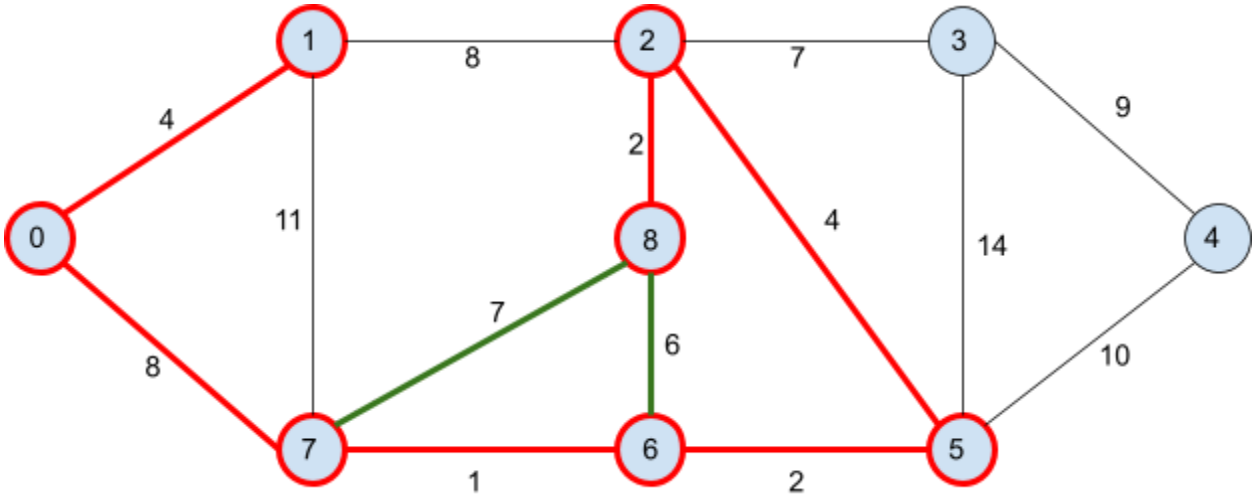
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), **(0-7)**
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5),** (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 8
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

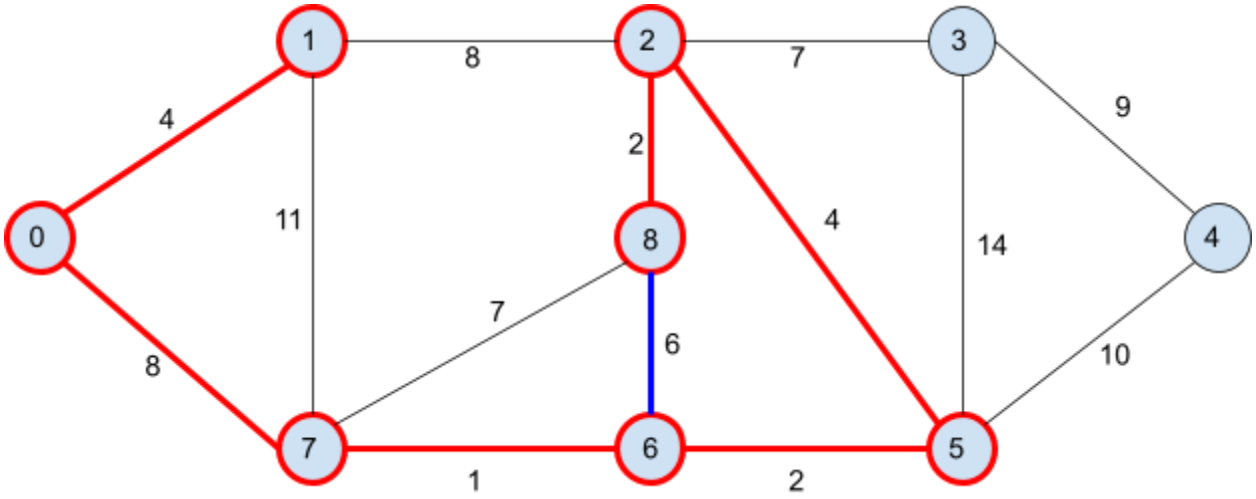
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), **(0-7)**
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5), (6-8)**
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 8
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

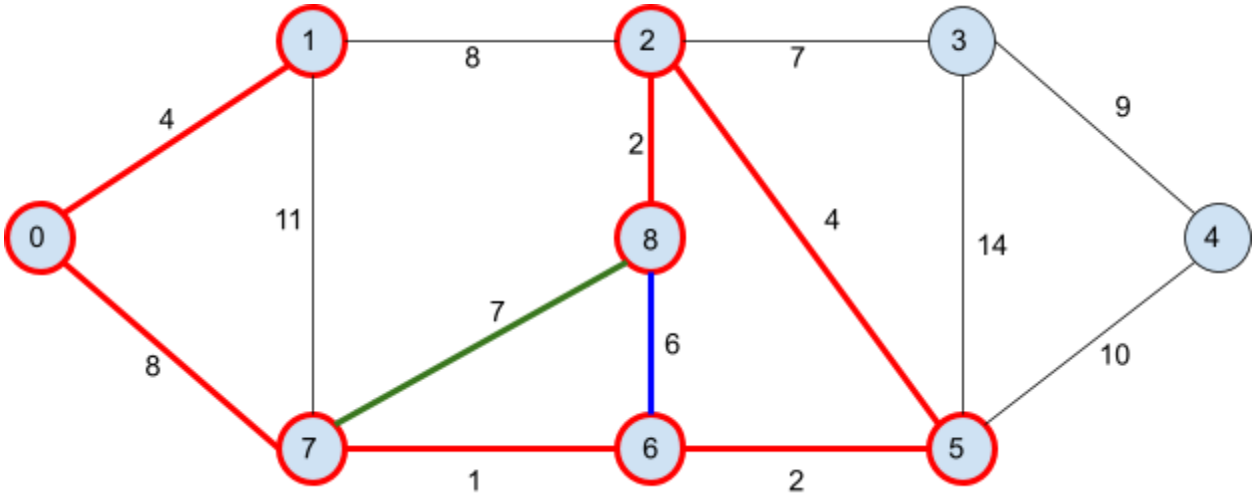
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 6
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

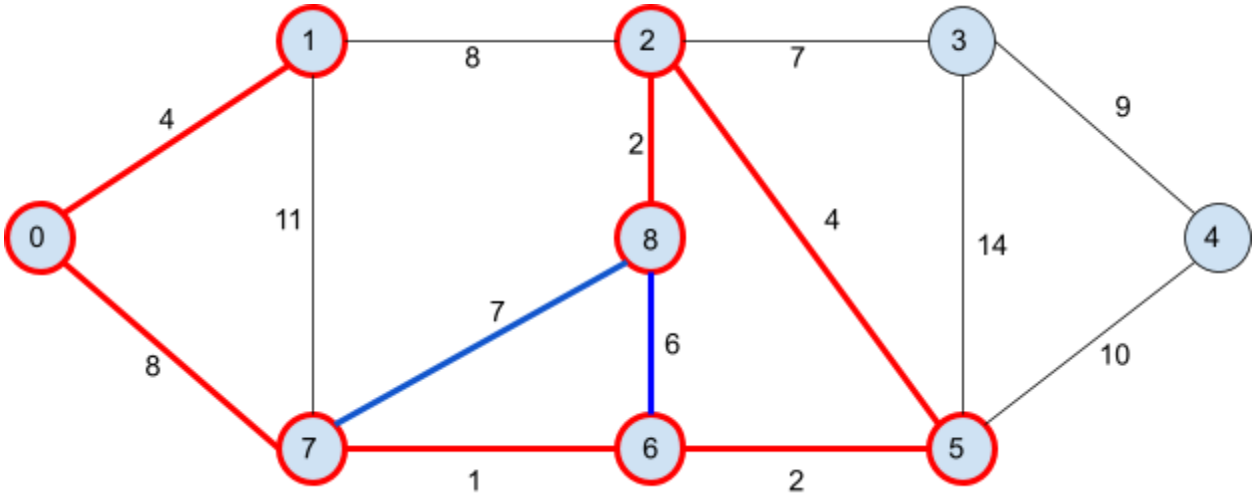
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 8
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

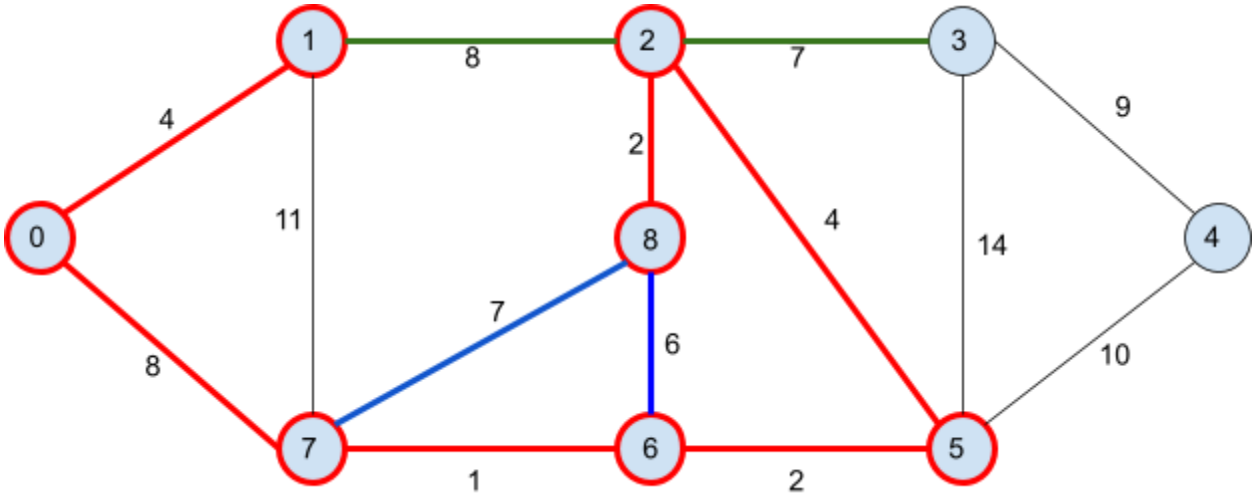
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 7
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

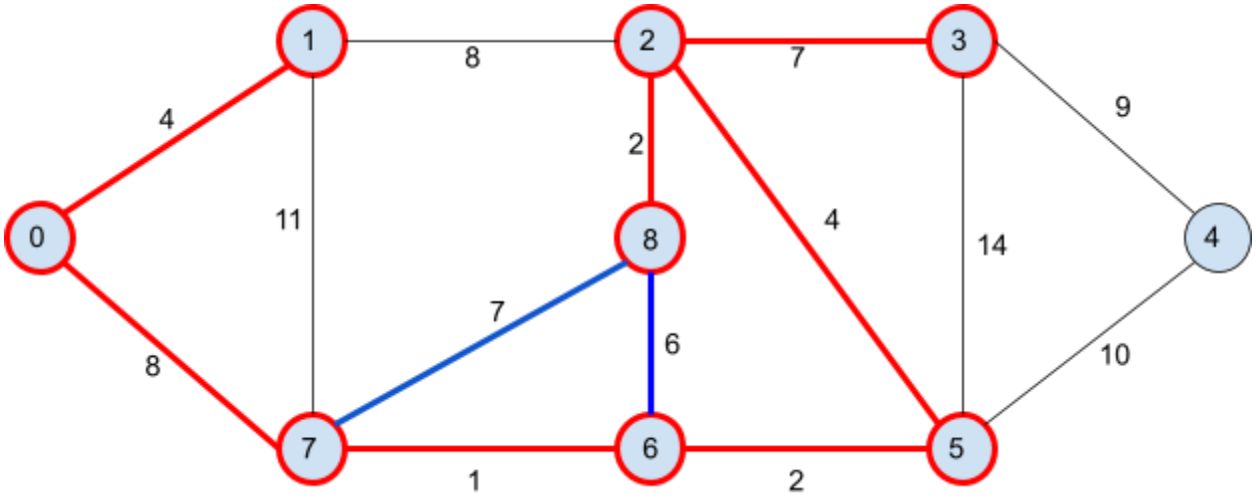
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), **(1-2)**, (1-7)
2: (2-8), (2-5), **(2-3), (2-1)**
3: **(3-2)**, (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), **(6-5), (6-8)**
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 2
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, 3, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8}

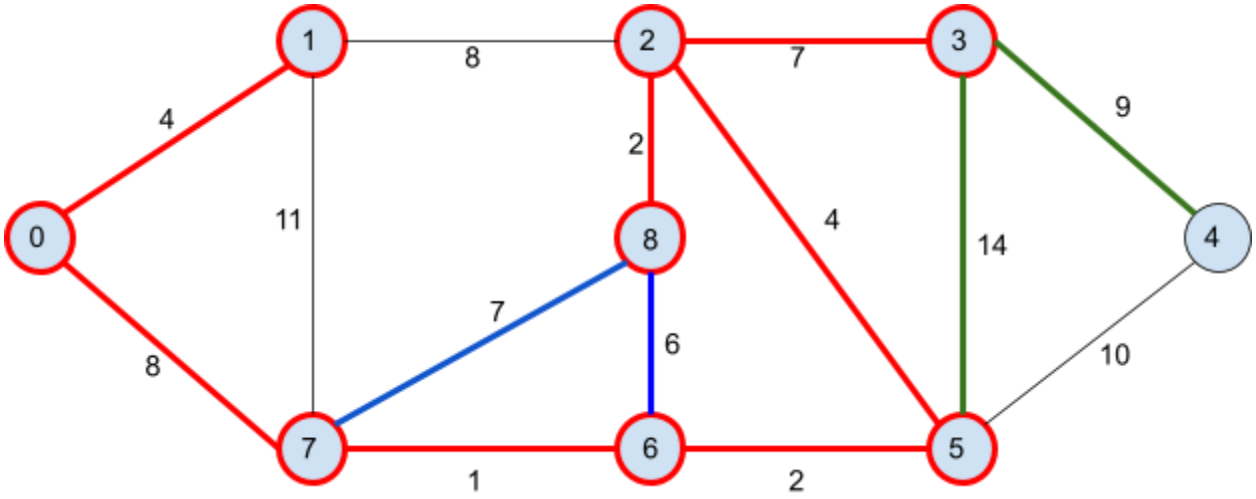
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 3
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3}

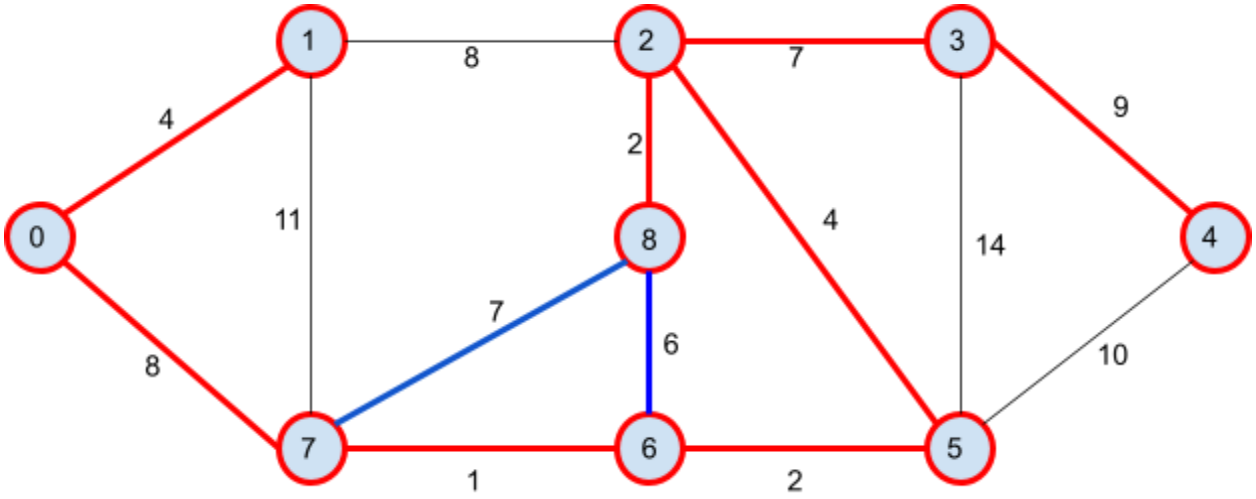
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: **(4-3)**, (4-5)
5: (5-6), (5-2), (5-4), **(5-3)**
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 3
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, 4, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3}

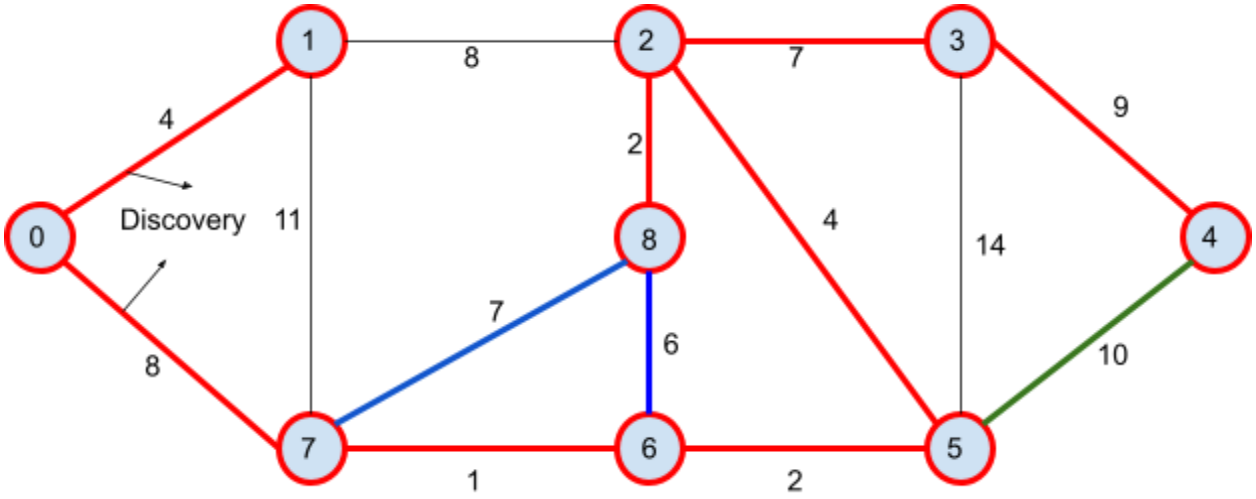
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 4
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

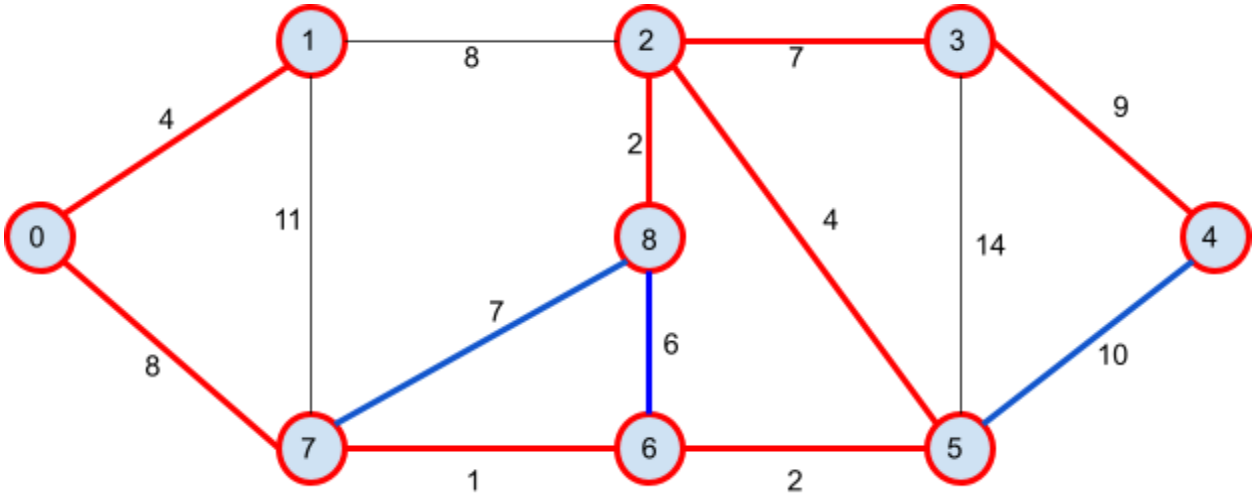
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 4
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

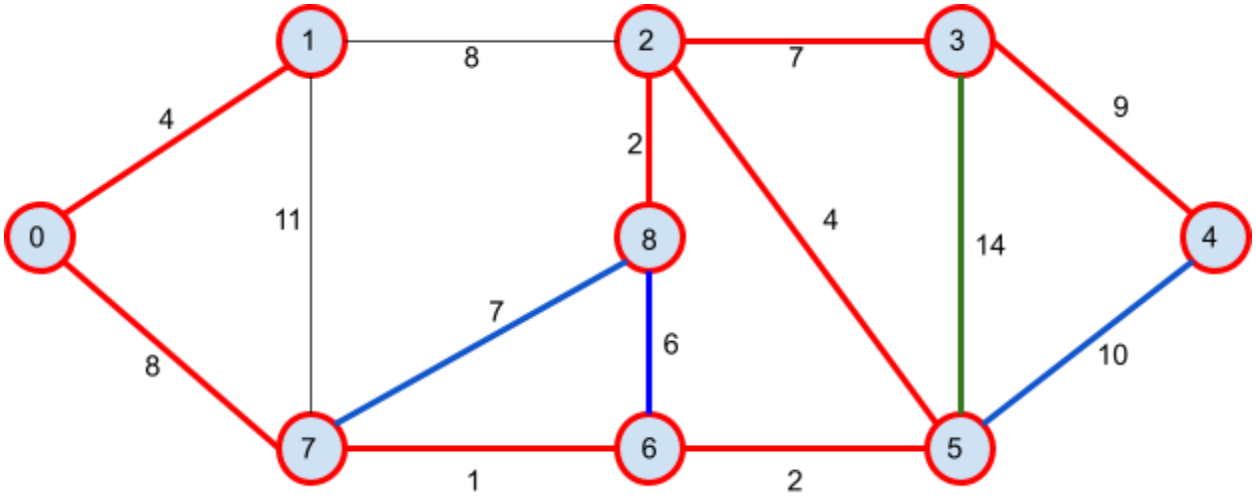
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 5
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

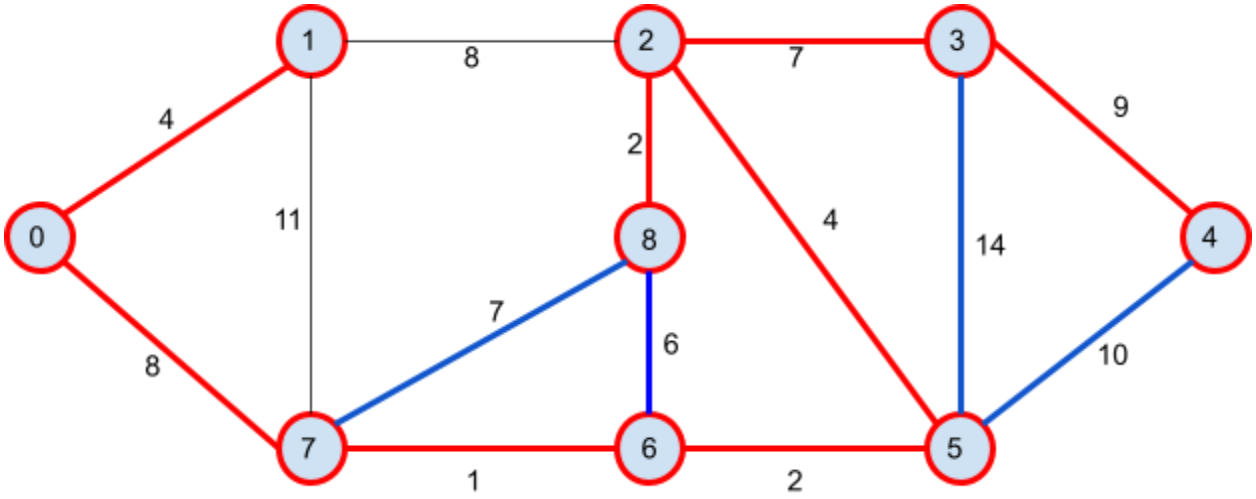
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 3
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

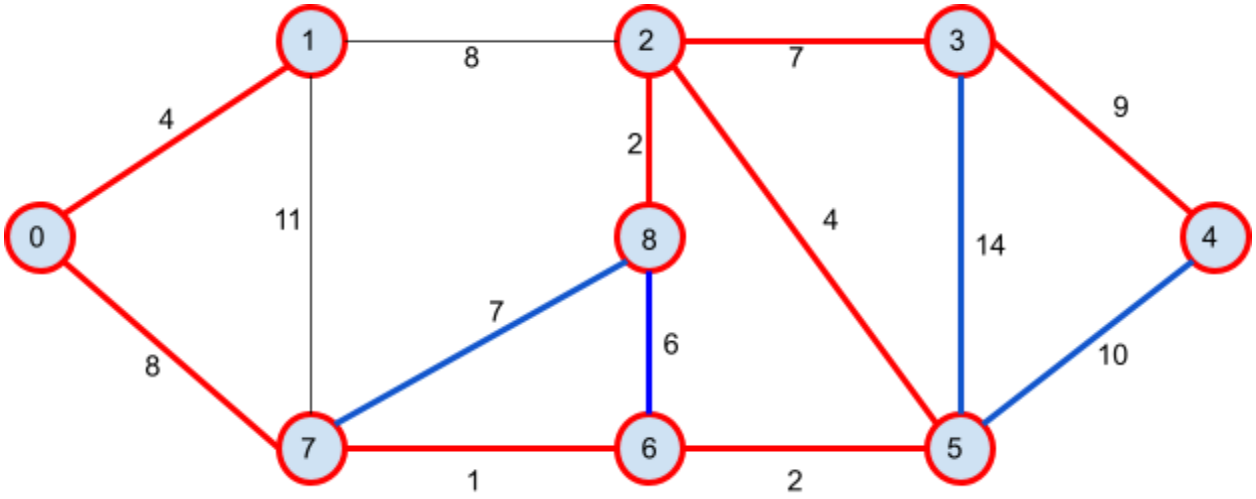
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 3
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

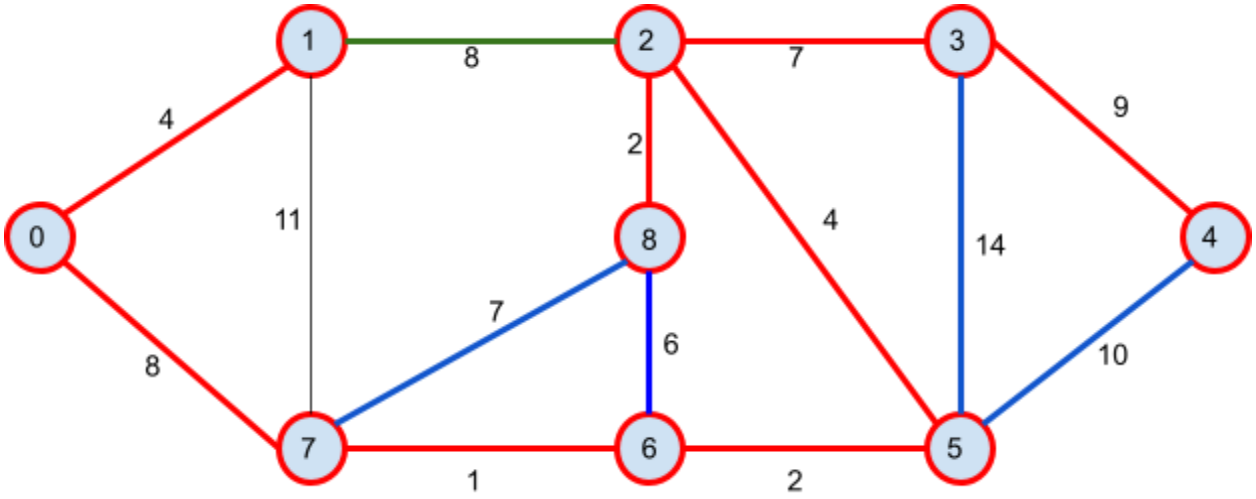
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 8
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

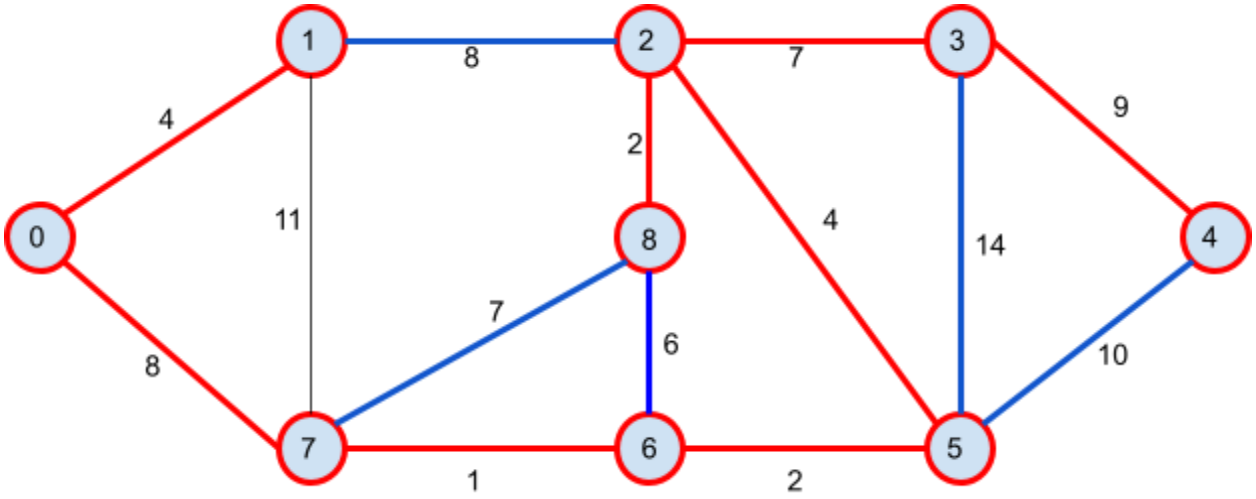
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), **(1-2)**, (1-7)
2: (2-8), (2-5), (2-3), **(2-1)**
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), **(5-4)**, **(5-3)**
6: (6-7), **(6-5), (6-8)**
7: (7-6), (7-8), **(7-0)**, (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 2
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

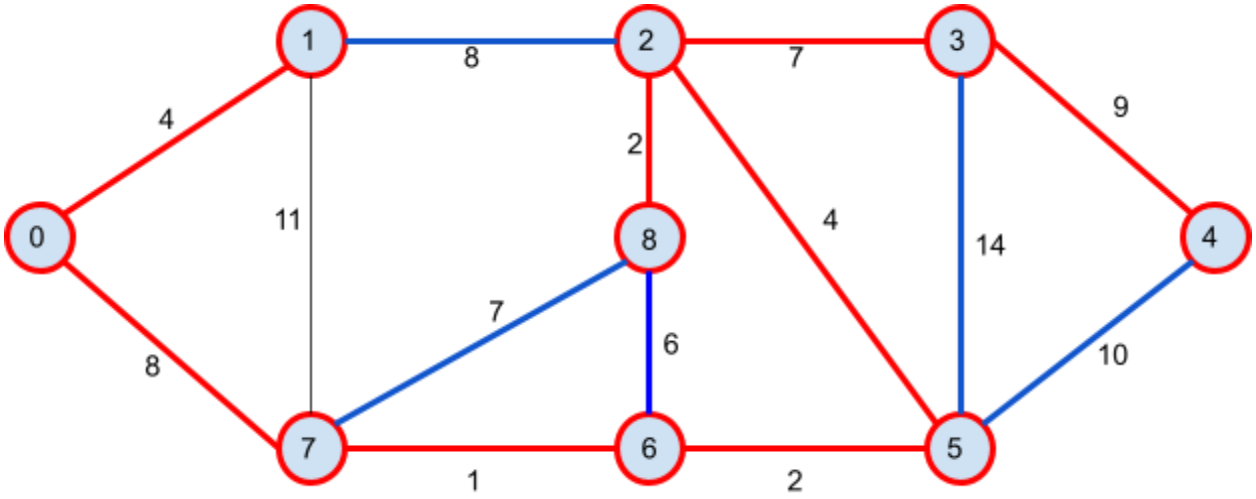
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 1
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

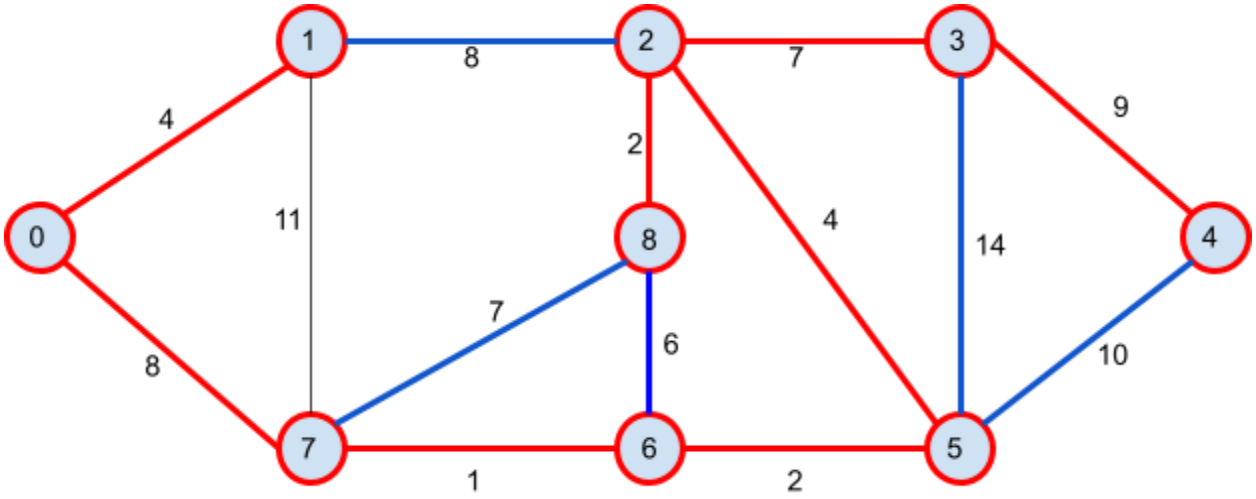
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 5
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

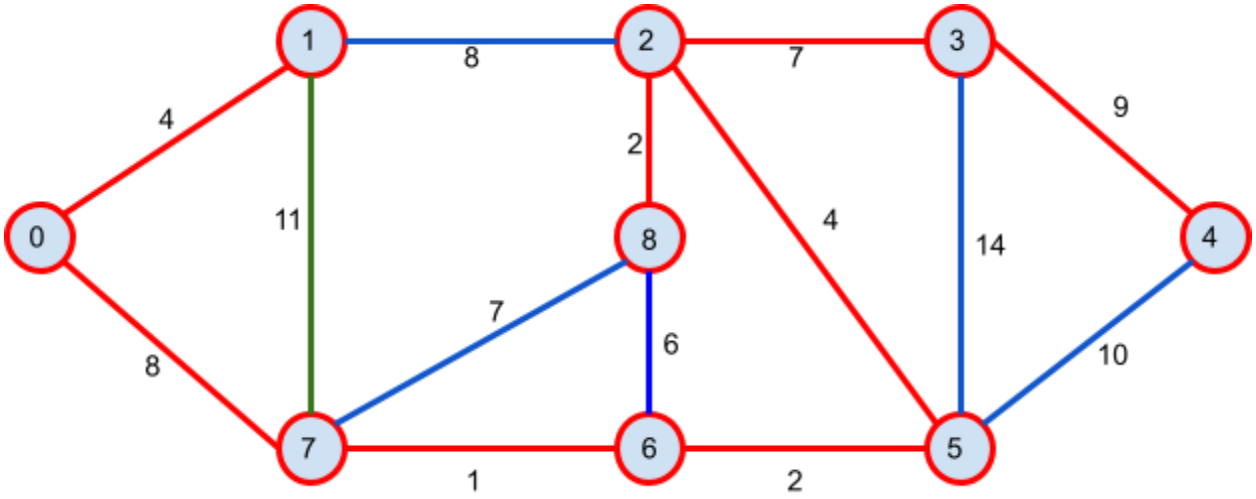
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 6
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

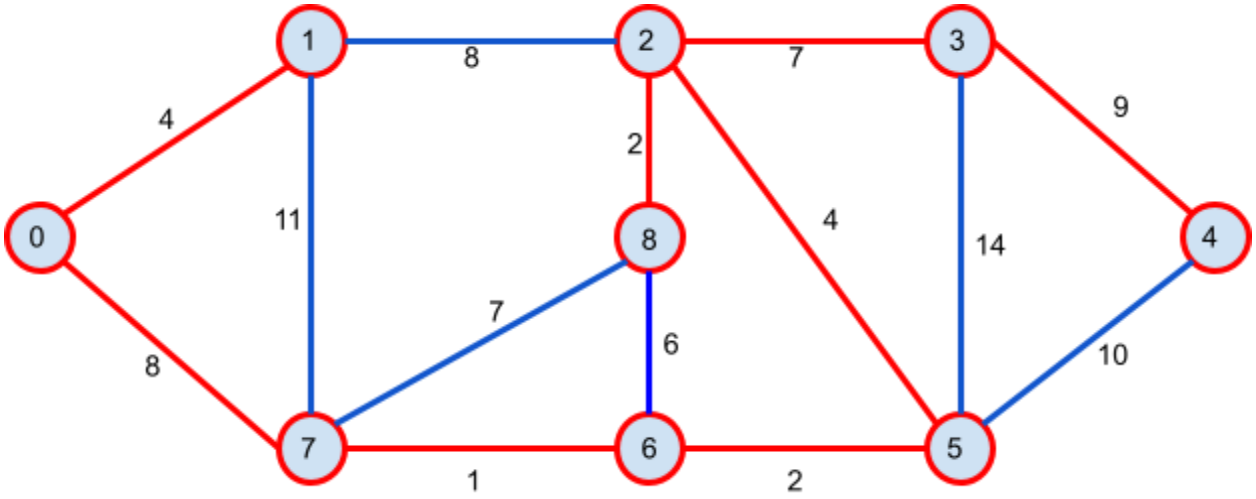
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 7
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

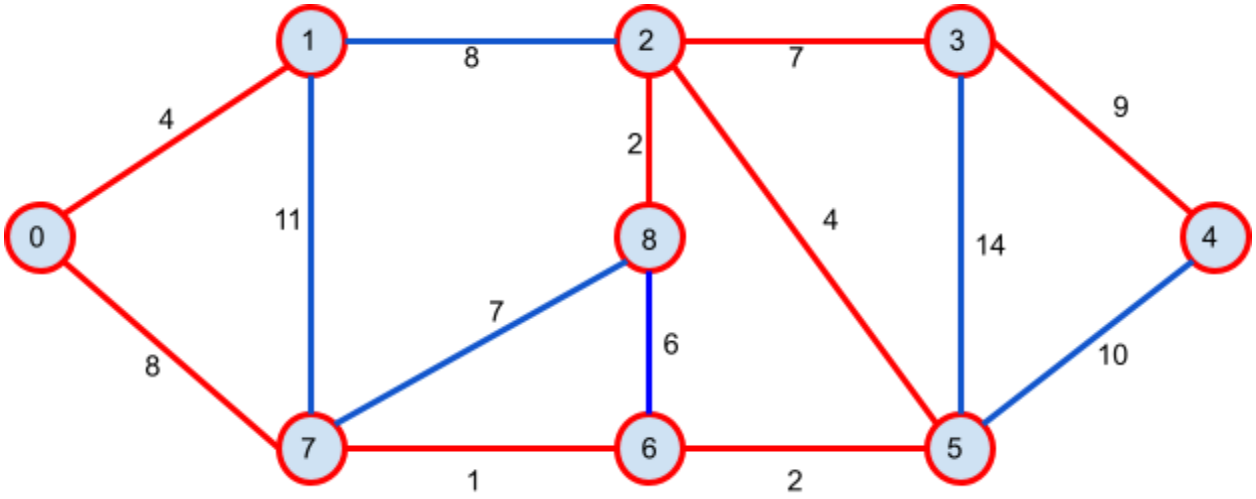
Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)
0: (0-1), (0-7)
1: (1-0), (1-2), (1-7)
2: (2-8), (2-5), (2-3), (2-1)
3: (3-2), (3-4), (3-5)
4: (4-3), (4-5)
5: (5-6), (5-2), (5-4), (5-3)
6: (6-7), (6-5), (6-8)
7: (7-6), (7-8), (7-0), (7-1)
8: (8-2), (8-6), (8-7)

Current Node: 1
Unvisited Nodes: {~~0~~, ~~1~~, ~~2~~, ~~3~~, ~~4~~, ~~5~~, ~~6~~, ~~7~~, ~~8~~}
Visited Nodes (in order): {1, 0, 7, 6, 5, 2, 8, 3, 4}

Red vertex => discovered vertex
Red edge => discovered edge
Black edge => undiscovered edge
Green edges => edges to analyze for finding shortest path from current node/vertex (based on edge weight) (Green vertex pairs in table indicate those edges)
Blue edge => Back edge (leading to an already discovered vertex)



Vertex (edge-adjacent vertex)	Visited Vertices (In Order)	**Discovery Edges**	**Back Edges**
0: (0-1), (0-7)	**1**	**-**	**-**
1: (1-0), (1-2), (1-7)	**0**	**(1-0)**	**-**
2: (2-8), (2-5), (2-3), (2-1)	**7**	**(0-7)**	**(7-1)**
3: (3-2), (3-4), (3-5)	**6**	**(7-6)**	**-**
4: (4-3), (4-5)	**5**	**(6-5)**	**-**
5: (5-6), (5-2), (5-4), (5-3)	**2**	**(5-2)**	**(2-1)**
6: (6-7), (6-5), (6-8)	**8**	**(2-8)**	**(8-6), (8-7)**
7: (7-6), (7-8), (7-0), (7-1)	**3**	**(2-3)**	**(3-5)**
8: (8-2), (8-6), (8-7)	**4**	**(3-4)**	**(4-5)**

Discovery Edges (in bold red) => structured in the manner of the current node connected to the next discovery node, where the reverse direction if this node is also counted and crossed out from the first column on the left.

Back Edges (in bold blue) => the edge from the current node to be determined to be a back edge for the current node.

All of these definitions are shown at each step where necessary.