Documentation – lab 2

1. Write a program that, given a directed graph and two vertices, finds a lowest length path between them, by using a forward breadth-first search from the starting vertex.

For graph1k:

Path between 1 and 100: 6

Path between 100 and 1: 5

For graph10k:

Path between 1 and 100: 8

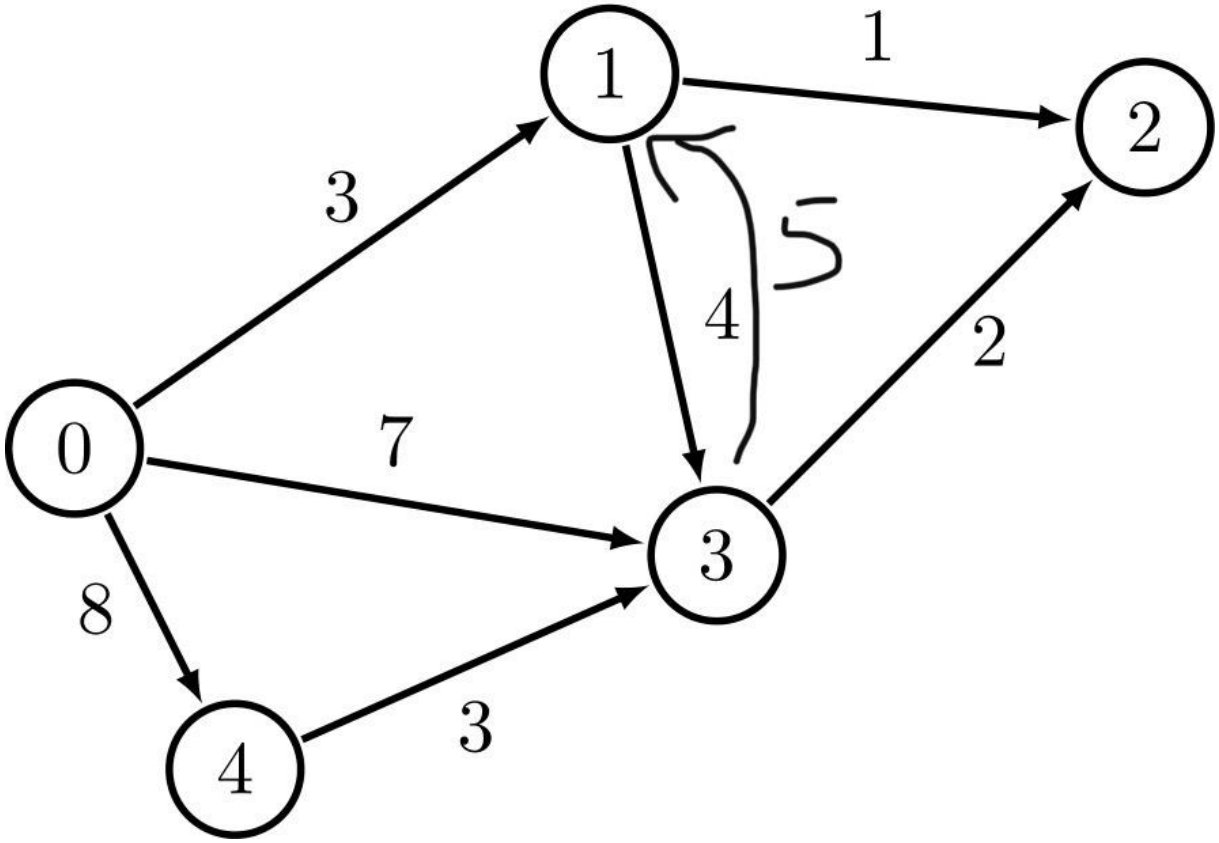
Path between 100 and 1: 7

For graph100k:

Path between 1 and 100: 8

Path between 100 and 1: 8

Graph:



For the given graph:

1. Distance between 0 and 2:

Step1:

Visited = [T, F, F, F, F]

Queue = [0]

Dist = [0, 9999999, 9999999, 9999999, 9999999]

Pred = [-1, -1, -1, -1, -1]

Step2:

Dist = [0, 1, 9999999, 1, 1]
Queue = [1, 3, 4]
Visited = [T, T, F, T, T]
Pred = [-1, 0, -1, 0, 0]

Step3:

Dist = [0, 1, 2, 1, 1]
Visited = [T, T, T, T, T]
Pred = [-1, 0, 1, 0, 0]
Stop: Path = 2

2. Distance between 3 and 0:

Step 1:

Visited = [F, F, F, T, F]
Queue = [3]
Dist = [9999999, 9999999, 9999999, 0, 9999999]
Pred = [-1, -1, -1, -1, -1]

Step 2:

Visited = [F, T, T, T, F]
Queue = [1, 2]
Dist = [9999999, 1, 1, 0, 9999999]
Pred = [-1, 3, 3, -1, -1]

Step 3:

Visited = [F, T, T, T, F]
Queue = [2]
Dist = [9999999, 1, 1, 0, 9999999]
Pred = [-1, 3, 3, -1, -1]

Step 4:

Visited = [F, T, T, T, F]
Queue = []
Dist = [9999999, 1, 1, 0, 9999999]
Pred = [-1, 3, 3, -1, -1]

The queue is empty so there is no path between 3 and 0.