

```

startVertex = self.getVertex()
endVertex = self.getVertex()
if (graph.isVertex(startVertex) != True or graph.isVertex(endVertex) !=
True):
    print("Invalid vertices.")
else:
    q = PriorityQueue()
    prev = {}
    dist = {}
    q.put((startVertex, 0)) # 2nd argument is priority
    dist[startVertex] = 0
    found = False
    while not q.empty() and not found:
        x = q.get()[0] # dequeues the element with min value of priority
        #print(x)
        for y in graph.dictOut[x]:
            if y not in dist.keys() or dist[x]+graph.getCost((x,y))<dist[y]:
                # in case y hasnt been parsed yet, or the new found road is
cheaper
                # dist[y] gets the new cost, x is the new predecessor of y
                # we add y and it's new distance to the priorityQ
                dist[y] = dist[x] + graph.getCost((x, y))
                q.put((y, dist[y]))
                prev[y] = x
            if x == endVertex:
                found = True
    print(dist)
    print(prev)
    print(f"The distance from {startVertex} to {endVertex}
is:{dist[endVertex]}!!!")

```