```
def shortest_path(graph, start, end):
    """
    This function creates and uses 3 dictionaries (dist,next and visited), one for retaining the path chosen, one for distances to each vertex from the ending vertex and one for checking the visited vertices
    :param graph: graph object
    :param start: vertex from the graph
    :param end: from the graph
    :return: lowest length path between start and end, and its length
    """
    visited = {}
    dist = {}
    next = {}

    for i in graph.inbound.keys():
        visited[i] = False

    for i in graph.inbound.keys():
        dist[i] = float('inf')

    for i in graph.inbound.keys():
        next[i] = None

    dist[end] = 0
    visited[end] = True

    queue = [end]
```

```
while queue:
    current_vertex = queue.pop(0)
    for i in graph.inbound[current_vertex]:
        if not visited[i]:
            visited[i] = True
            dist[i] = dist[current_vertex] + 1
            next[i] = current_vertex
            queue.append(i)

if dist[start] == float('inf'):
    print("There is no path")

path = []
current = start
while current is not None:
    path.append(current)
    current = next[current]

return path, dist[start]
```