

* Shortest Path Problem

- Same approach as BFS for queue. just we will keep track of path distance and backtrack as soon as we find the short path. We will use pointers to the tracking so that book-keeping is maintained.

- General Algorithm

while queue:

current node = queue.popleft()

if current node == end:

path[]

while current node $\neq \phi$

path.insert(0, current node)

current node = parent.get(current node)

return path

for neighbour in graph[current node]

if neighbour not in visited

queue.append(neighbour)

visited.add(neighbour)

parent[neighbour] = current node

return [] // no path exists.

- Complexity Analysis

We are using two arrays which are both traversed only once. Hence we will run the program in linear time.

$$O(m+n)$$