GTU DEPARTMENT OF COMPUTER ENGINEERING

CSE-222 SPRING 2023 HOMEWORK REPORT

ONUR ATASEVER

210104004087

Analysis Of BreadthFirstSearch:

I assumed that n is the number of vertices and m is number of adjacent in graph.

For loop to fill visited and path arrays is T(n) = $\Theta(n)$

For best case first vertex's first edjacent is target. So It is T(n) = $\Theta(1)$

But since there is $\Theta(n)$ before over all running time is T(n) = $\Theta(1) + \Theta(n) = \Theta(n)$

For worst case It traverses for all vertex's and all adjacent's of vertices.

So T(n, m) = $\Theta(n*m)$

Analysis Of Dijkstra Algorithm:

For loop to fill vertices set, distance and pred arrays is T(n) = $\Theta(n)$

While loop is $\Theta(n)$ time. Removing an element from set is $\Theta(\log n)$ time. And for loop which is in while is $\Theta(m)$ time since it traverses on adjacents of vertex. And to find minimum distance in vertex set is $\Theta(n)$.

So T(n) = $\Theta(n) * \Theta(\max(\log n, m, n))$

Running Times(for nano seconds):

Map01:

BFS: 477823000

Dijkstra: 65341520500

Map02:

BFS: 294188200

Dijkstra: 53264142400

Map03:

BFS: 353732000

Dijkstra: 40349797500

Map04:

BFS: 256747400

Dijkstra: 71790665700

Map05:

BFS: 208517800

Dijkstra: 38287930900

Map06:

BFS: 280783500

Dijkstra: 36719837700

Map07:

BFS: 286846100

Dijkstra: 51366080300

Map08:

BFS: 288375800

Dijkstra: 42768080300

Map09:

BFS: 417968000

Dijkstra: 47680204200

Map10:

BFS: 157455100

Dijkstra: 36878761000

tokyo:

BFS: 1739531600

Dijkstra: 377977294900

vatican:

BFS: 2284334300

Dijkstra: 395987465200

pisa:

BFS: 1909378900

Dijkstra: 365898355400

triumph:

BFS: 1781002100

Dijkstra: 375774629800