**Breadth First Search**

Pseudo Code:

BFS(M,S,E) // M is map, S is start node, E is end node

Let Q be a queue

Q.enqueu(S)

While(Q is not empty)

N = Q.dequeue()

set N to visited

if(N is E)

Create Path()

return

else

foreach (neighbor node in N)

if (node is not visited)

Q.enqueue(node)

set node to visited

Properties:

* Complexity:
  + Time Complexity: O(N) where N is the number of Nodes in the map
  + Space Complexity: O(N) where N is the number on Nodes in the map
* Completeness: It will find the goal state if one exists.
* Admissibility: Since the cost for moving from one tile to another is uniformly 1, it will find a best solution.
* Irrevocability: Tentative