Devan Patel
CS 435 - Project 2 - Part 2

4a. DAG - Directed Acyclic Graph
- no cycles (no circular dependencies)
- edges are directed from one vertex to another and the existance of an edge between two nodes does not imply bi-directionality.
- the new graph will be guaranteed to have no cycles (unlike the prev graph).
- In the new graph, a vertex A may have an edge from itself to some vertex B but have no vertex back to itself directly from B.

5a. Dijkstra's algorithm is suited only for graphs with non-negative edge weights. It works on unweighted graphs as well (we can consider all weights to be the same). If src and dst are unconnected, Dijkstra's will not give a ssp or return False. Cycles are also permitted. Our graph is connected (path exists from any node A to some other node B) and it is also weighted. Hence, we can use Dijkstra's algorithm on our graph.

6c. Heuristic: Manhattan Distance = h
It is admissible because it never overestimates the cost from some point (x_1, y_1) to (x_2, y_2).
It is consistent because for every node i and j: $h(i) \leq$ ~~cost~~ $g(i,j) + h(j)$.
$g(i,j)$ represents the cost from i to j.
In the given problem statement, the Manhattan Distance represents the shortest path if all ~~neighbours~~ adj. vertices are connected on the grid.

7a. Dijkstra's finalizes all nodes in any given graph (assuming the graph is ~~D.A.G~~ connected and has no negative edge weights). A* finalizes a small fraction of nodes because the heuristic used is both admissible and consistent.