4.b)

Manhattan heuristics is admissible, because all nodes need to move to the correct position between itself and the destination, thus it never overestimate. Also, Manhattan heuristics is consistent because the distance between two nodes are always the same and the distance always less or equal to the distance from any neighbor plus the cost of reaching neighbor.

Euclidean heuristics is admissible, because all nodes need to move to the correct position between itself and the destination in a straight line. Thus, it never overestimates. In Minkowski distance formula, when p=1 it is Manhattan distance, when p=2, it is euclidean distance. Therefore, since the distance is always the same and the distance is always less or equal to the distance from any neighbor plus the cost of reaching neighbor. Therefore, euclidean heuristics is consistent.

Random heuristics is not admissible, because it always chooses a random number, therefore it is not grantee will have the best distance, therefore, it could overestimate. Since random heuristics use a random number, it is not consistent. Because it uses a random number, there is no grantee that the distance is always less or equal to the distance plus the cost of reaching goal.

Base on the compiling:
Manhattan: Search nodes expanded: Path found with total cost of 210 in 0.1 seconds
Search nodes expanded: 549
Euclidean:Search nodes expanded: Path found with total cost of 210 in 0.1 seconds
Search nodes expanded: 557
Random: Search nodes expanded: Path found with total cost of 210 in 0.1 seconds
Search nodes expanded: 629

Base on the node expansion above, Manhattan heuristics did the lowest expansion, euclidean expand slightly higher than Manhattan. Random expansion
is the biggest. Therefore the performance is Manhattan<Euclidean<Random.

5.

BFS:Path found with total cost of 54 in 0.1 seconds

Search nodes expanded: 682

DFS:Path found with total cost of 298 in 0.0 seconds

Search nodes expanded: 576

Uniform:Path found with total cost of 54 in 0.1 seconds

Search nodes expanded: 682

A*:Path found with total cost of 54 in 0.1 seconds

Search nodes expanded: 682([SearchAgent] using function astar and heuristic nullHeuristic)

Base on the compiling result above, the total cost and search nodes expand   for BFS, uniform and Astar are the same, because when there is no food and ghost(food and ghost will increase the cost of getting destination) in the map, it only need to find the optimal path to go to the final destination.

For DFS, it has the best performance in running time, but it have the highest cost, because it checks all the possible paths.