Report AI Homework n.1

Manuel Del Verme 1769408

1.1) Several algorithms were tested:

Local search algorithms:
failed to find a solution on most of the runs, I suspect this is because of the iteration limit and sparseness of acceptable solutions (92 out of 8^8 states) while local minima are common over the whole space in the whole search space.

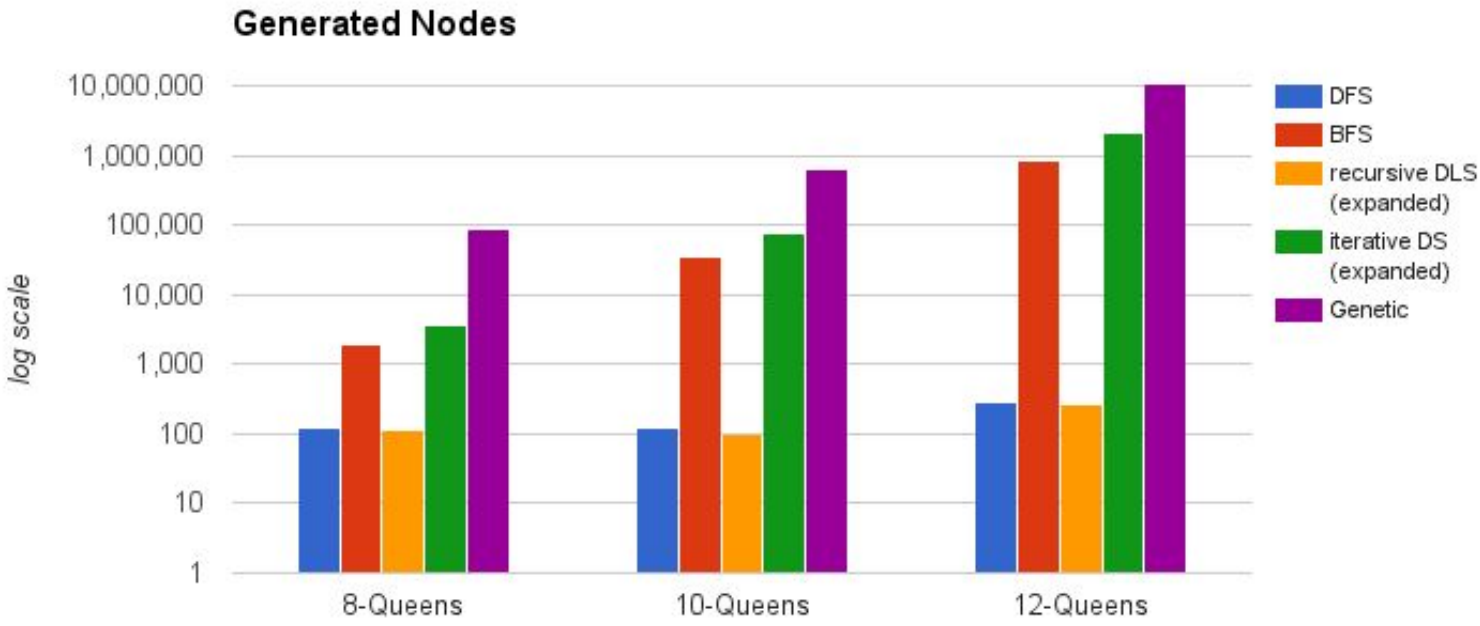
The Hill climbing and Simulated annealing are very likely to converge to a local minimum due to the topology of the search space.

The genetic algorithm suffers of the same problem, converging to a local minima and taking a long time to evolve out of it, thus generating solution of orders of magnitude slower.

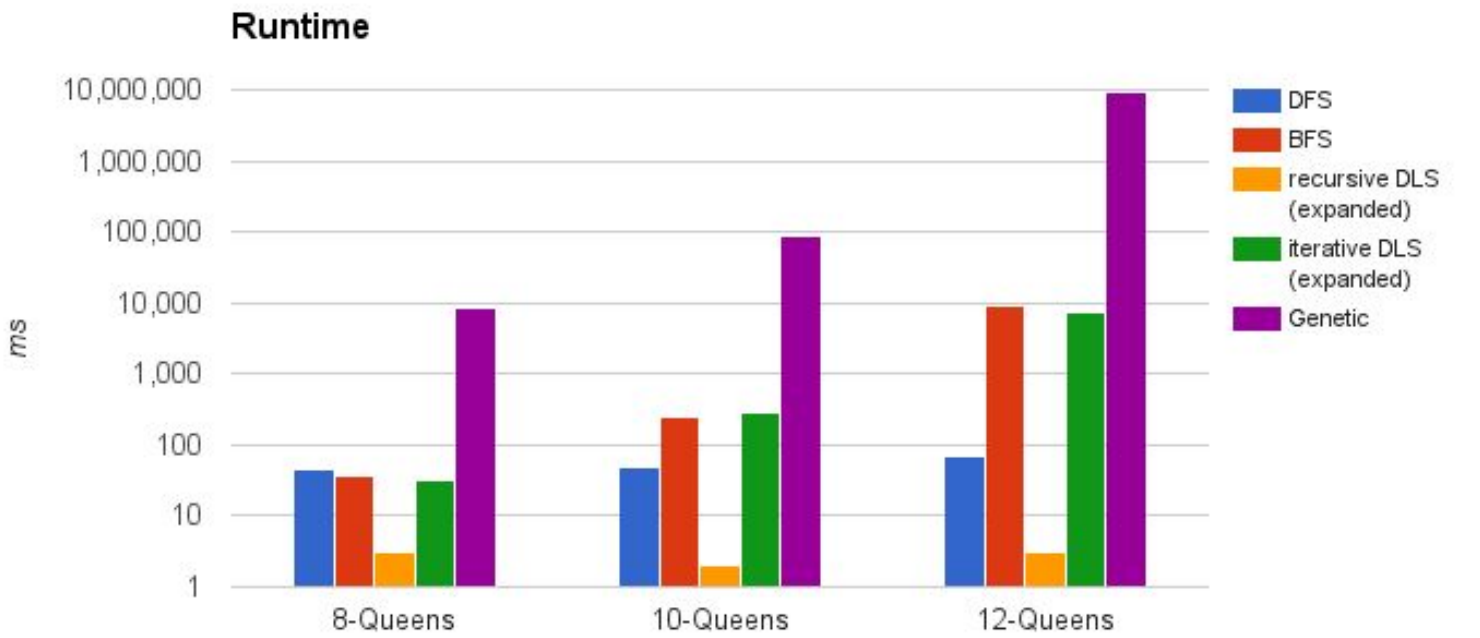
Graph search algorithms:

The problem has a branching factor of O(n^2): N rows, N columns.

Considering that the only valid solutions have a depth of N, where N is the number of queens, equal to the maximum depth of the search space, it's not surprising to see depth based algorithms performing better than breadth based ones, since the search space can be interpreted as a tree with solution on the leaves, a breadth based algorithm has to explore the whole tree except for the leaves before seeing any solution.

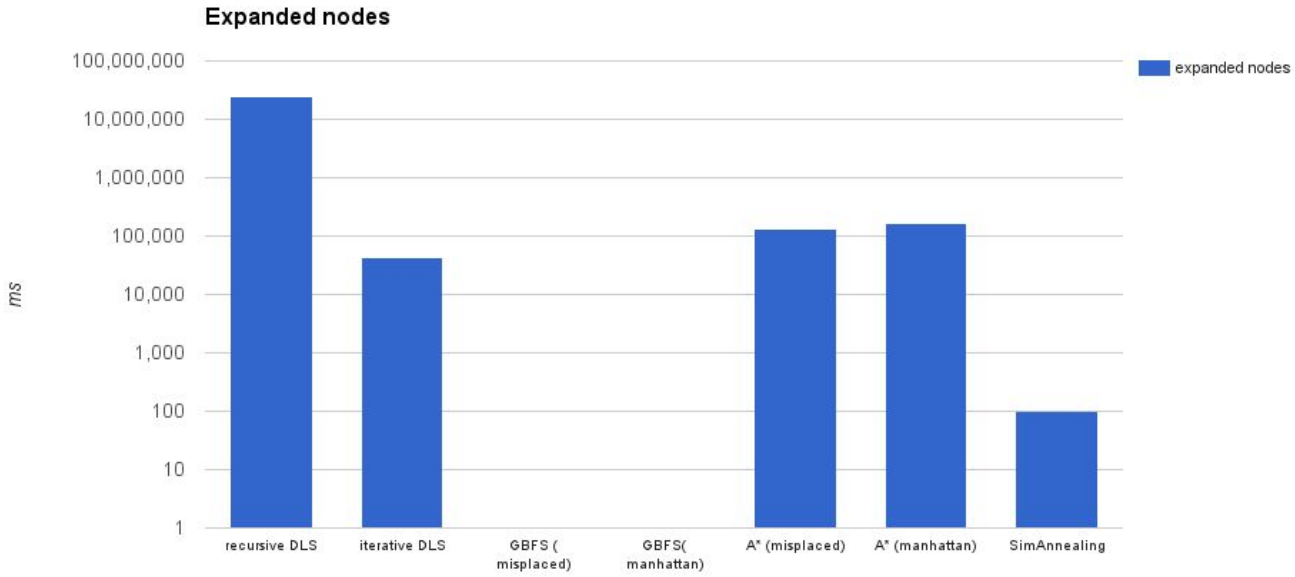


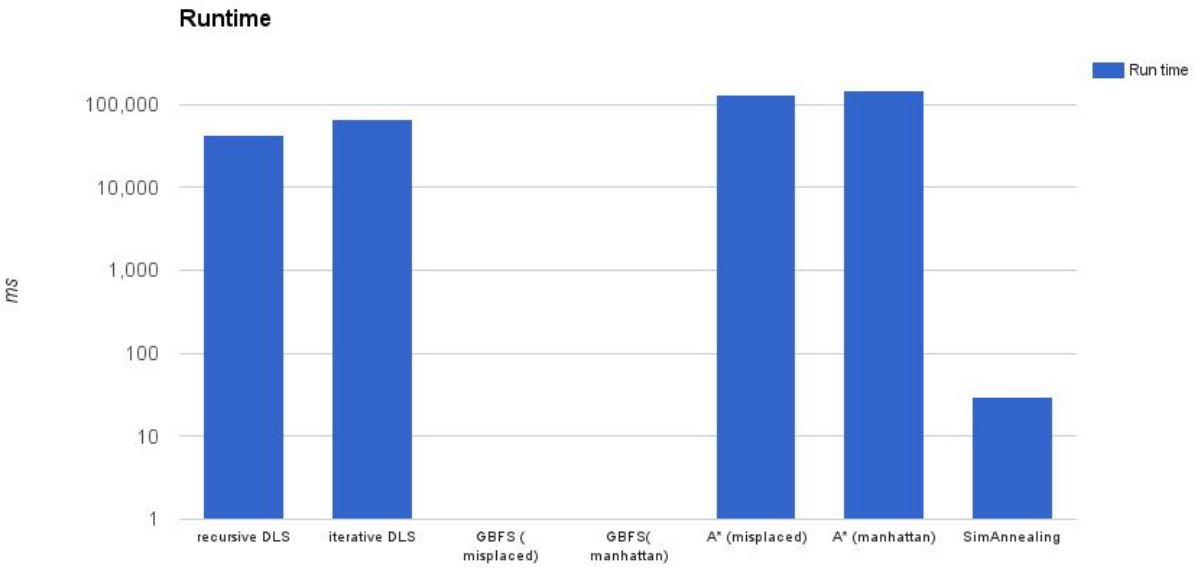
Depth based algorithms explore 1 order of magnitude less nodes than breadth based algorithms because of the high branching factor.



Since the time required to explore a node is negligible the runtime is proportional to the number of explored nodes.

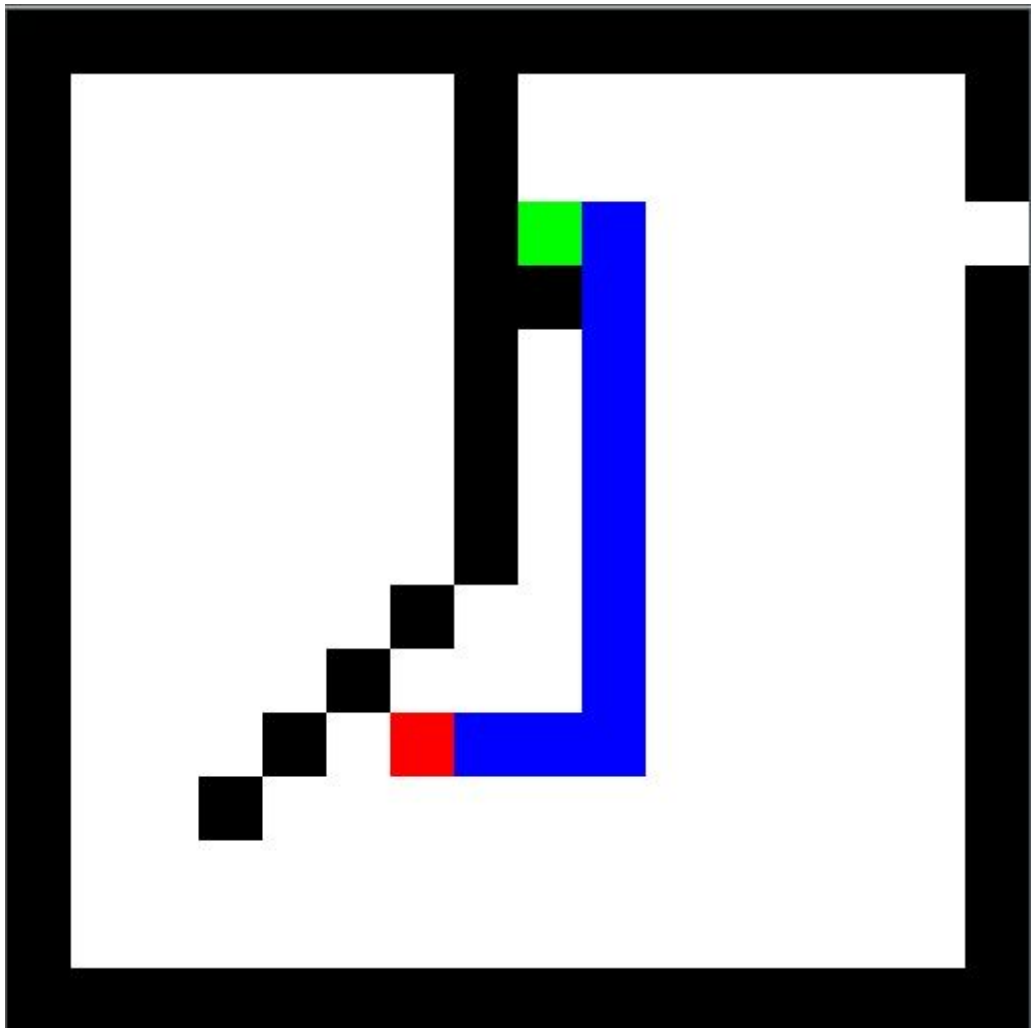
1.2) The 16 Puzzle has a much lower branching factor (4 for up, down, right, left), but an higher depth.
Simulated Annealing failed to find a solution over 15 experiments.
Greedy Best First Search algorithms failed to find a solution after 30min or 1.8e+6ms, and were interrupted.





During A* the computation of the heuristic function creates a cost per node expanded, slowing down the algorithm

2.1) IDS



IDS took 1386 ms to find a simplified version of the problem, having v=0, since we know the goal location an informed search will greatly outperform this algorithm.

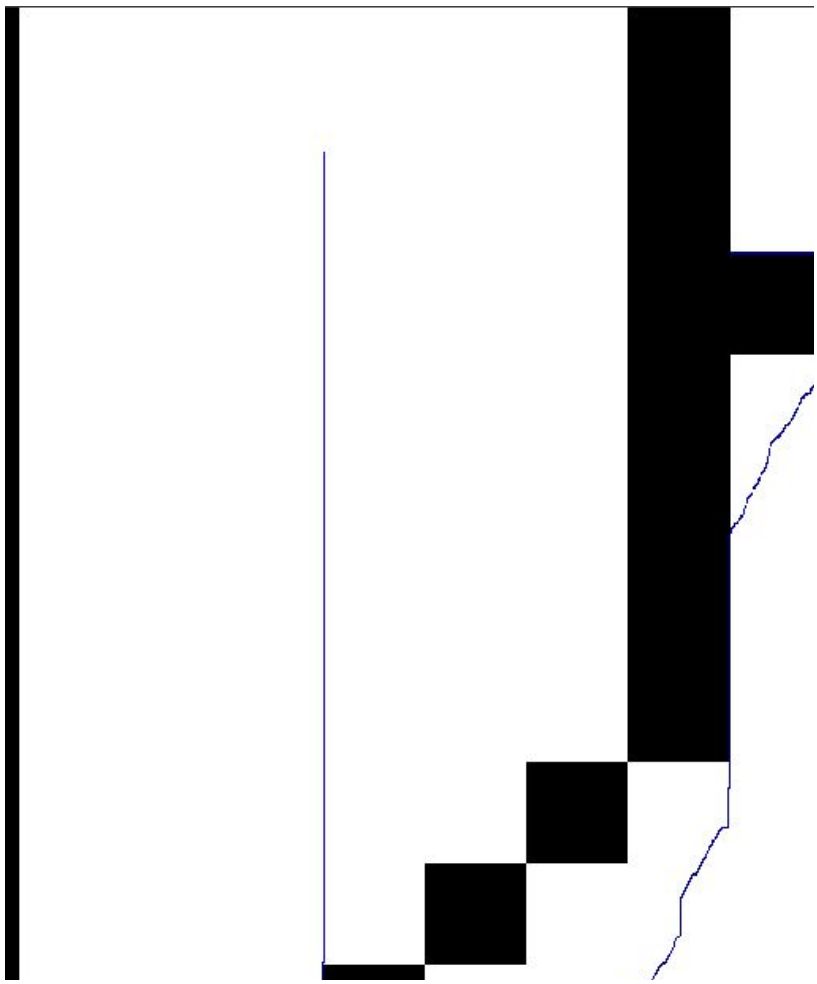
2.2) A*
The chosen heuristic function was in the form:
H = ε * manhattan_distance(robot, goal).

The more the heuristic function underestimates (low ε) the real cost the fastest the solution is found in worlds with a clear path, as the optimal path diverges from a straight line a pessimistic heuristic (ε = 1) becomes faster, non-ammissibile heuristics (ε > 1) can be even faster but fail to find the optimal solution.

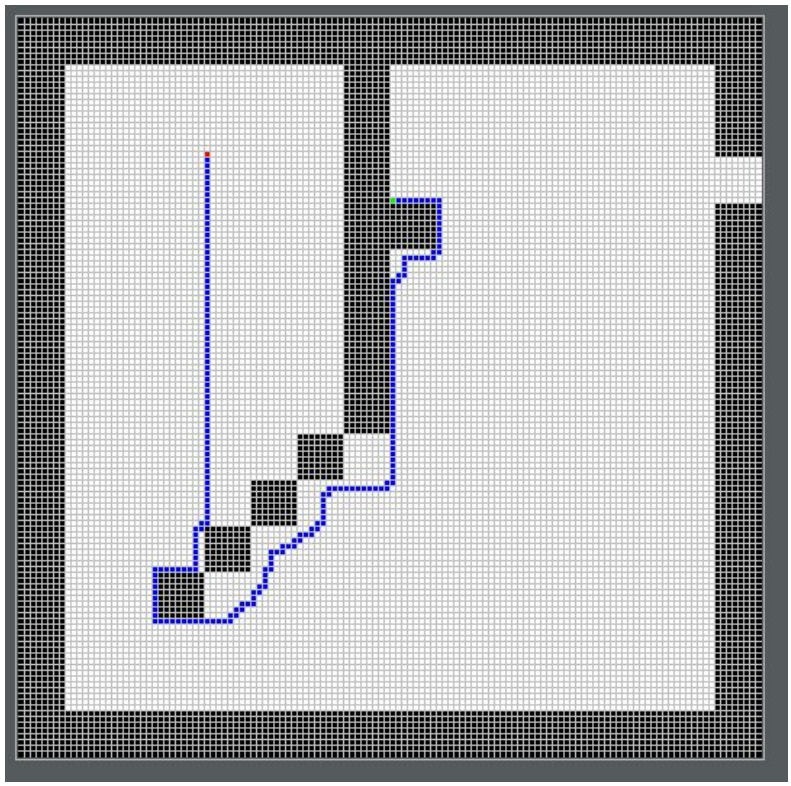
Values of v and runtime:

v	ms
3	156
6	2940
7	6679
8	44077

For values of V >9 memory becomes the limiting factor.



V = 8



V = 3