Assignment 2 - Analysis of the Scatter Plots

- Reminder, all the result and files mention below are inside submission files and can be regenerated by running `main.py`

1. The use of a heuristic function decrease the number of nodes need to expand to find a solution.
   - For more detail, check number of nodes generated by Dijkstra, Bi-Bs, A^*, Bi-A^*, and MM for each a test instance shown in `expansion_stat.csv`
2. Option d $\rightarrow$ The running time will decrease, but not as much as the number of expansions decreases. From looking at the statistical time in `profile.csv`, the runtime of calling `heappush()` and `heappop()` function, which pushs and pops a state from and into the open list, decreases approximately in the same proportion in which number of expansions decreases. However, there is additional runtime cost need to be considered. The most noticeable is that calling `heapify()` function to heapify the heap when a value in the heap is updated. From the statistical time in `profile.csv`, algorithms that employs heuristic function drastically increase the number of times of heapifying the open list compared to the algorithms that do not use a heuristic function. Since heapify a heap is costly, which is $O(n)$, along with the drastic increase in the number of heapifying the open list, the runtime will not decrease as much as the number of expansions decreases. Notice that `heapify()` need to call comparison operators or functions to compare each element in the heap structure, those comparison operators or functions will produce additional runtime cost as well. Another factor is that cost of arithmetic on heuristic function, particularly, when the heuristic function require operations other than simple addition and subtraction, and need to be called frequently since the algorithm is heuristic.

3. From the number of nodes generated by MM and Bi-A^* shown in `expansion_stat.csv`, MM tends to perform fewer expansions than Bi-A^* in general except that there are few test instances where MM perform more expansions or nearly the same expansions as Bi-A^*.

4. From the number of nodes generated by MM and Bi-A^* shown in `expansion_stat.csv`, the heuristic-guided bidirectional algorithms does not deliver their promise of substantially reducing the number of expansions one needs to perform to solve a problem. The substantially saving requires a situation of which the search space substantially large, or worst case, b^d where b is the cost of the optimal solution, $C*$, in order to take affects.

5. The distribution of the scatter plot show that $A*$ and MM have similar number of expansions. In some specific cases, MM has more expansions than A^*. Specifically, there are few cases that MM has substantially more expansions than A^*. Thus, MM does not offer substantially reduce in the number of expansions to A^* via having two searches in different direction. In some specific cases, MM performs worse than A^*.

   For cases that MM has substantially more expansions than A^*, since MM inherits the characteristics of bidirectional searche and employs heuristic additionally, it suffers the same issue bidirectional search has when encounter a search wit no solution path. In order to terminate a bidirectional search in a search with no solution path, bidirectional search need to exhaust all nodes in both open list compared to single direction search which only need to exhaust all nodes in only one open list. This means that bidirectional search expands more and thus takes longer time to conclude that a search has no solution path, compared to single directional search. Employing heuristic make the difference more obvious. On a single directional search, heuristic guide the search and prevent the search from deviating away the general direction toward the goal. Thus, a single directional search will stay in a optimal direction toward the goal without unnecessary expansions. This significant reduce number of expansion from without heuristic to with heuristic.

Aside the cases of which searches have no solution path, despite the two directional searches in MM use heuristic as a single directional search does, there's no significant different between MM and A^*. Since a single directional search like $A*$ employ heuristic, it unlikely to suffer from issues uninformed single directional serach (Dijkstra) does as long as heuristic is consistent and admissible. Additionally, MM need to modifiy slight different heuristic and terminating condition to ensure both direction do not expands too far while meet in the same node as well as obtain an optimal solution. Overall MM only offer substantically saving in reduce of number of expansion when the search space is in $O(C^*^d)