**Problem 4.2.1**

Both algorithms belong to the class of "best-first search" algorithms, which are those that can use both the information discovered thus far while exploring the search space, denoted by g(n), and a heuristic function, denoted by h(n), which calculates the distance to the goal node for each node n in the search space (often represented as a graph).

For each node n in the graph (or search space), indicated by f, each of these search methods defines an "evaluation function" (n). This evaluation function is used to identify the node that is "expanded" first during a search or the node that is initially taken off the "fringe" (or "frontier" or "border") to "visit" its offspring.

The evaluation function for the A\* method is f(n)=g(n)+h(n), where h is a valid heuristic function. With a time complexity of O(bm) and a potential space complexity of a polynomial, a greedy search is incomplete and suboptimal. Complete, ideal, and having an O-level time and space complexity, A\* (bm). Consequently, A\* requires more memory overall than the greedy BFS. When there is a large search space, A\* is no longer useful. A\*, however, also ensures that the algorithm will finally come to an end and that the path identified between the beginning node and the destination node is the best one. On the other hand, greedy search requires less memory but does not offer assurance A\*'s of optimality and completeness. Therefore, the "best" algorithm will vary depending on the situation, although both use best-first searches.

**Problem 4.2.1**

* If the heuristic function h(n) is always less than or equal to the actual cheapest cost from n to the objective, then it is said to be admissible if h(n) never exceeds h\*(n).
* If A\* employs an allowable heuristic and h(goal) = 0, then it is admissible.
* A\* will always identify the best solution if the heuristic function, h, consistently underestimate the real cost (h(n) is less than h\*(n)).