This lecture discusses two algorithms for graph traversal: greedy search and A* search, using the example of finding a route from Arad to Bucharest on a map of Romania. Greedy search employs a heuristic, specifically the straight-line distance to Bucharest, to guide the search process. The heuristic is an estimate and should be a lower bound of the actual cost. The algorithm uses a priority queue to always expand the most promising node based on the heuristic, without considering the actual cost until the end. The lecture illustrates this with an example where the greedy search starts from Arad and explores neighboring cities based on their heuristic values.

The greedy search algorithm is demonstrated by pushing nodes into a priority queue based on their heuristic values. For instance, from Arad, the algorithm considers neighbors like Timisoara, Sibiu, and Zerind, and selects the one with the lowest heuristic value. This process continues until Bucharest is reached, resulting in a path that is not necessarily the shortest but is found quickly without backtracking. The example path found by greedy search is Arad, Sibiu, Fagaras, and Bucharest, with a total cost of 417 km.

The lecture then introduces the A* search algorithm, which combines both the heuristic and the actual cost to find the shortest path. A* search also uses a priority queue but sorts nodes based on the sum of the actual cost and the heuristic estimate. The algorithm keeps track of the path and the cost to reach each node, ensuring that the shortest path is found. The example path found by A* search is Arad, Sibiu, Rimnicu Vilcea, Pitesti, and Bucharest, with a total cost of 415 km.

A* search is guaranteed to find the shortest path as long as the heuristic does not overestimate the actual cost. The lecture explains that even with a heuristic of zero, A* search would still find the shortest path, similar to Dijkstra's algorithm, but would explore more nodes. The advantage of greedy search is its low memory cost and fast direction towards the target, while A* search has a higher memory cost but guarantees the shortest path.

In conclusion, the lecture compares the two algorithms, highlighting that greedy search is efficient with a good heuristic but not guaranteed to find the best solution, whereas A* search, with its higher memory cost, ensures the shortest path. The lecture provides a comprehensive understanding of how these algorithms work and their applications in graph traversal, particularly in the context of finding routes on a map.