Exhaustive Search

When considering the processing of a graph $G = (V, E)$, one problem to be solved is searching the graph for a particular vertex of that graph having certain characteristics. Often such a search amounts to "walking" the graph along various edges and constructing various paths through the graph until the desired vertex is found. Within this context, we will consider various algorithms for performing such a search.

The initial set of graph search algorithms we will consider are referred to as "exhaustive" because, in the limit, they may require searching the set of all possible vertices. When evaluating any search strategy, but especially exhaustive search strategies, we are interested in evaluating a number of different performance criteria. Of particular interest are the following:

1. Completeness: A "complete" search strategy is one that guarantees finding a target vertex or path given such a target vertex or path exists.

2. Time Complexity: As with any other algorithm, "time complexity" refers to a measure of the amount of work required by the execution of an algorithm.

3. Space Complexity: Similar to time complexity, "space complexity" refers to a measure of the amount of memory required by the execution of an algorithm.

4. Optimality: When searching for some vertex or path, a cost measure can be applied to the search process and the cost of the path by which the vertex is found determined. This provides a measure of the cost of that path. An optimal search algorithm finds the path to the element that minimizes cost. Optimality is sometimes referred to as "admissibility."

We now consider two specific exhaustive search methods—breadth-first search and depth-first search.

- Breadth first search

- Depth first search