Graphs and paths

You are given the following data definitions for a graph

```
;; A Node is a Symbol
;; INTERP: represents the name of a node in a graph

;; A Distance is a PosInt
;; INTERP: represents distance in miles

;; An Edge is (list Node Distance Node)
;; e.g. (list 'A 10 'B)
;; INTERP: represents an edge from 'A to 'B with the distance from 'A to 'B
being
;;         10 miles

;; A Path is a [List-of Edge]
;; A Graph is a [Set-of Edge]
;; NOTE: a Set is an unordered List where there are no duplicates
```

You are asked to provide the following functions on Graphs

1. `valid-path?` that consumes a graph and a path, and returns true if the path is valid for the graph, and false otherwise. A path is valid for a graph if it is possible to follow each edge in order from the path on the graph, i.e., the graph contains these edges from the path and we can follow them in the order given by the path.
2. `valid-st-path?` that consumes a graph `g`, a start node `s`, an end node `t`, and a path `p`, and returns true if starting at node `s` in `g` and following, in order, the edges in `p`, will lead us to t. The function should return false otherwise.
3. `find-st-path` that consumes a graph `g`, a start node s and an end node t and returns a path `p` that starts at `s` and ends at `t` in the graph `g`. The function should return false if there is no such path.
4. `find-shortest-distance-st-path` that consumes a graph `g`, a start node `s` and an end node `t` and returns the path `p` that starts at `s` ends at `t` in the graph `g` and when we add the distances of each edge is `p` that is the shortest distance from `s` to `t` in `g`. The function should return false if there is no such path.