

For #4 (ski-o problem) —

- Be sure to address how to handle controls (controls are associated with track segments, not junctions) and how to ensure that they are visited in the correct order.
- Design graphs, not algorithms! Don't modify some standard graph algorithm to handle controls and their sequencing, set up the graph and your usage of the standard graph algorithm instead.
- Embedded vs topological. This can be a bit more complicated than just either-or. A map is inherently embedded because it depicts things with real-world locations and their relative positions on the map carry essential meaning for users navigating from one point to another on the map, but not all applications require all of that information — a graph can be topological even when its origins are in an embedding. In this case, once the graph is built, does the fact that there can be a specific embedding for the graph matter for solving the problem?