# Day 8

You have a list of airline tickets represented by pairs of departure and arrival airports [from, to]. Your task is to reconstruct the itinerary in the correct order. It's important to note that all of the tickets in the list belong to a man who initially departs from airport A. Therefore, the itinerary should start with airport A.

## Examples

```
findPath([["C", "F"], ["A", "C"], ["I", "Z"], ["F", "I"]]) → ["A", "C", "F", "I", "Z"]

findPath([["A","C"],["A","B"],["C","B"],["B","A"],["B","C"]]) → ["A","B","A","C","B","C"]
// Another possible reconstruction is ["A","C","B","A","B","C"].
// But it is larger in lexical order.

findPath([["Y", "L"], ["D", "A"],["A", "D"], ["R", "Y"], ["A", "R"]]) →  ["A", "D", "A",
 "R", "Y", "L"]
```

**Notes:**

- In case there are multiple valid itineraries, you should prioritize returning the itinerary that has the smallest lexical order when it is read as a single string. To illustrate, if we compare the itineraries ["A", "B"] and ["A", "C"], the former has a smaller lexical order.
- It can be assumed that every ticket in the list will contribute to at least one valid itinerary.
- It is crucial to use each ticket once and only once while constructing the itinerary.