Itinerary in C++

```
#include <iostream>
#include <unordered_map>
#include <string>

using namespace std;

int main() {
    unordered_map<string, string> map;
    map["Chennai"] = "Banglore";
    map["Bombay"] = "Delhi";
    map["Goa"] = "Chennai";
    map["Delhi"] = "Goa";

    // Create a hashmap to mark if a city is a potential
source
    unordered_map<string, bool> psrc;
    for (auto it = map.begin(); it != map.end(); ++it) {
        string src = it->first;
        string dest = it->second;

        psrc[dest] = false; // Destination city cannot be a
source
        if (psrc.find(src) == psrc.end()) {
            psrc[src] = true; // Source city if it is not a
destination in the map
        }
    }

    string src = "";
    for (auto it = psrc.begin(); it != psrc.end(); ++it) {
        if (it->second == true) {
            src = it->first;
            break;
        }
    }

    // Print the itinerary
    while (true) {
        if (map.find(src) != map.end()) {
            cout << src << " -> ";
            src = map[src];
        } else {
            cout << src << ". ";
            break;
        }
    }

    return 0;
}
```

Dry Run Example:

Input Data:

```
unordered_map<string, string> map;
map["Chennai"] = "Banglore";
map["Bombay"] = "Delhi";
map["Goa"] = "Chennai";
map["Delhi"] = "Goa";
```

1. **psrc Mapping**:
    - Initially, all cities are marked as potential sources (true).
    - Iterating over map:
        - "Chennai" is a source (because it's not in the destination list).
        - "Bombay" is a source.
        - "Goa" is a destination, so it's marked as false.
        - "Delhi" is a destination, so it's marked as false.
    - Final psrc will be:

      cpp
      Copy code
      psrc = { "Bombay" = true, "Delhi" = false, "Goa" = false, "Chennai" = false }

2. **Finding the Source City**:
    - The first city with true in psrc is "Bombay".
    - Set src = "Bombay".
3. **Building the Itinerary**:
    - Starting from "Bombay":
        - "Bombay" -> "Delhi"
        - "Delhi" -> "Goa"
        - "Goa" -> "Chennai"
        - "Chennai" -> "Banglore"
    - Print "Bombay -> Delhi -> Goa -> Chennai -> Banglore."

Output:

Bombay -> Delhi -> Goa -> Chennai -> Banglore.

Output:
Bombay -> Delhi -> Goa -> Chennai -> Banglore.