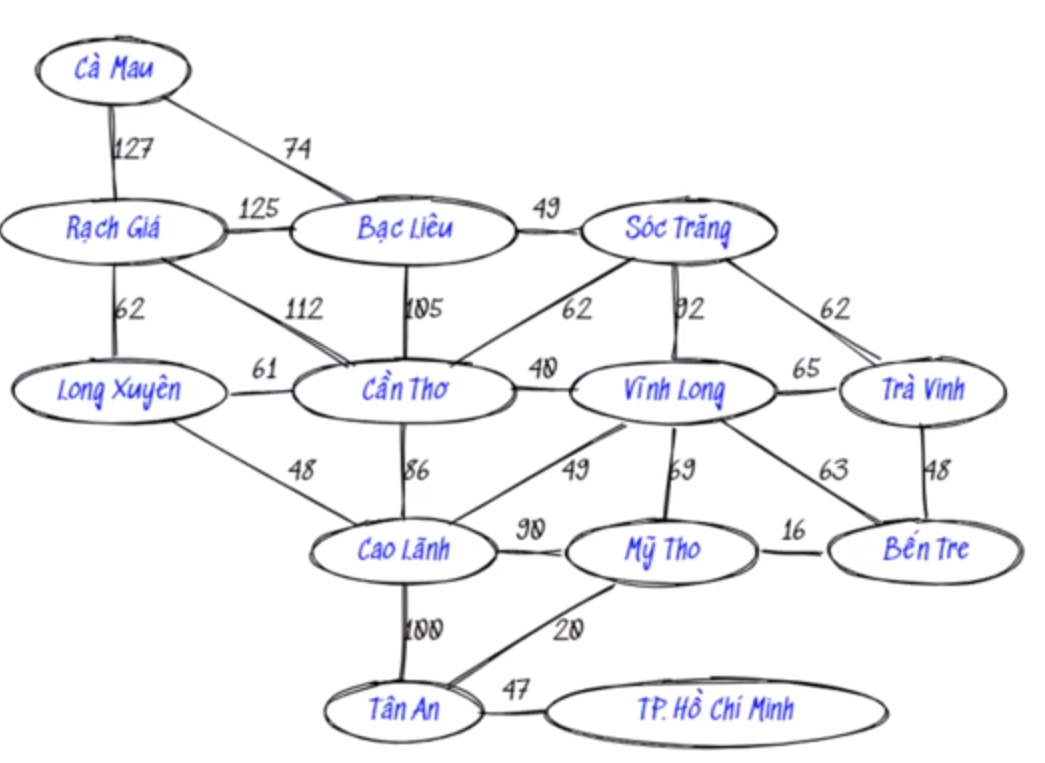
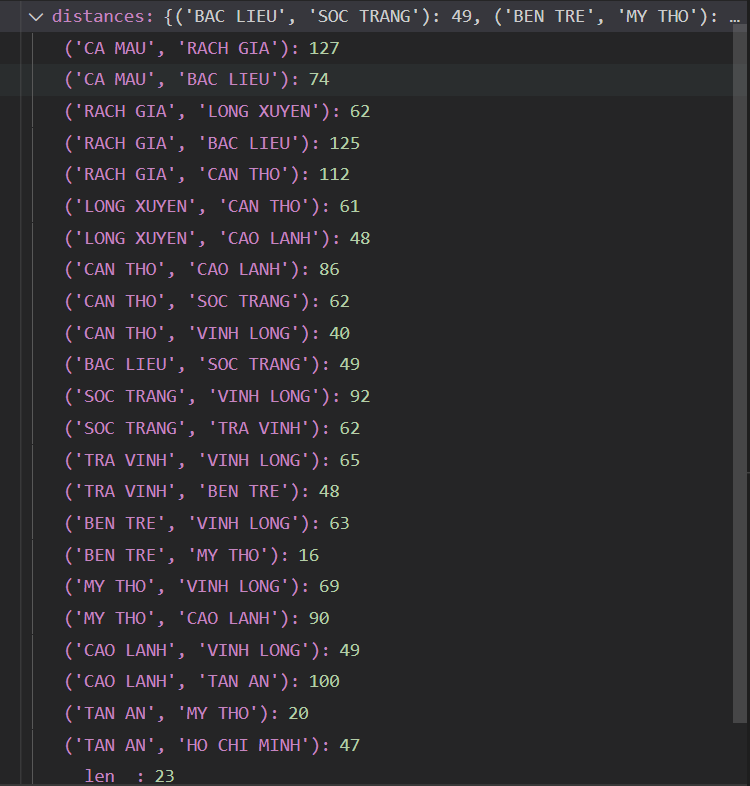
**Bài tập 1: BREADTH-FIRST SEARCH**

1. **Cách tạo cây**



* Giả sử ta có bản đồ như hình trên, để tạo ra bản đồ đầu tiên ta cần khai báo:
  + Tạo một biến để chứa class Graph()
  + Khai báo tất cả các node, trong trường hợp này có các node: 'CA MAU', 'RACH GIA', 'LONG XUYEN', 'BAC LIEU', 'CAN THO', 'SOC TRANG', 'VINH LONG', 'TRA VINH', 'CAO LANH', 'MY THO', 'BEN TRE', 'TAN AN', 'HO CHI MINH'
  + Cú pháp như sau:
    - Tên biến = Graph()
    - **Graph = Graph()**
* Sau khi đã khởi tạo class và khai báo số nodes có trong cây đó, tiếp đến ta tiến hành tạo cây để thuật toán Dijsktracó thể chạy được:
  + Sử dụng hàm **addEdge(FirstNode, SecondNode, Distance)** để tạo ra bản đồ. Ví dụ từ hình của bản đồ ta có thể add CA MAU và RACH GIA với khoảng cách là 127 (km)
    - **g.addEdge(‘CA MAU’, ‘RACH GIA’, 127)**
    - Tương tự ta tiếp tục tiến hành như thế đến khi tạo ra hết bản đồ
  + Hình dưới thể hiện bản đồ dưới dạng dictionary:



1. **In ra cách di chuyển của thuật toán Dijkstra:**

* Để có thể in ra cách mà thuật toán **Dijkstra** di chuyển, ta thực hiện gọi hàm **dijkstra(graph, Start)** ví dụ: **dijkstra(graph, ‘A’)**

1. **Bảng tổng hợp các hàm:**

|  |  |
| --- | --- |
| **Tên hàm và cú pháp** | **Công dụng** |
| Graph() | Khởi tạo cây |
| addEdge(FirstNode, SecondNode, Distance) | Tạo nhánh mới |
| dijkstra(graph, Start) | In ra cách thức di chuyển của thuật toán dijkstra từ node đã định sẵn trước |
| Shortest\_path(graph, Start, destination) | Tìm đường đi ngắn nhất dựa trên dijkstra |