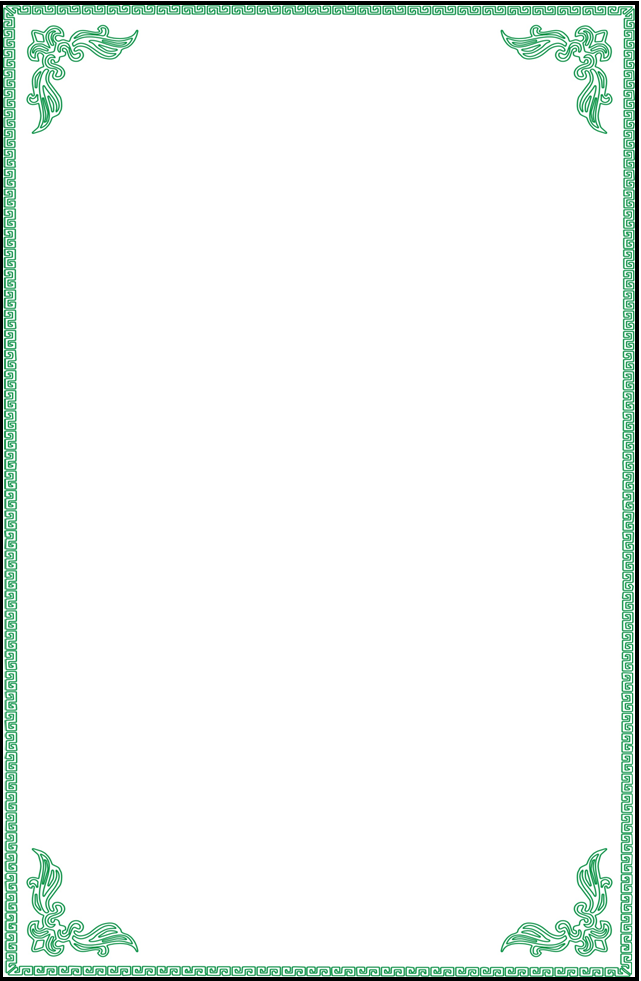
****TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**FLIGHT MAP**

Giảng viên hướng dẫn: **Phạm Trọng Nghĩa**

Sinh viên thực hiện: **Lai Gia Phú** **1712662**

Lớp: **17CTT5C**

TP. Hồ Chí Minh, tháng 1 năm 2019

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**FLIGHT MAP**

Giảng viên hướng dẫn: **Phạm Trọng Nghĩa**

Sinh viên thực hiện: **Lai Gia Phú** **1712662**

Lớp: **17CTT5C**

TP. Hồ Chí Minh, tháng 1 năm 2019

# Các thuật toán và cấu trúc dữ liệu sử dụng

Sử dụng vector<string> để lưu tên các thành phố theo thứ tự trong file cityFile.txt để lấy chỉ cột dòng đưa vào đồ thị.

Sử dụng struct ChiPhi chứa số hiệu, giá vé và thời gian bay của mỗi chuyến.

Dùng ma trận kiểu ChiPhi để lưu đồ thị được đọc ra từ file flightMap.txt.

Đọc file requestFile.txt để thực hiện tìm chuyến bay theo yêu cầu thông qua thuật toán Dijkstra kết hợp với đệ quy để tìm k đường đi tốt nhất.

# Giải thích phương thức và thuộc tính các hàm:

DocCityFile: dùng thư viện fstream để xử lý đọc file, getline từng dòng để lấy tên thành phố và push\_back vào vector<string>. Lúc này thì mỗi thành phố sẽ mang 1 giá trị vị trí i dùng để định vị trong đồ thị.

DocFlightFile: dùng thư viện fstream để xử lý đọc file, getline từng dòng để lấy tên 2 thành phố cùng với thông tin của chuyến bay, lưu vào ô tương ứng với vị trí 2 thành phố trong ma trận kiểu ChiPhi. Sử dụng thêm một số hàm trong thư viện string để tách chuỗi.

minDistance, minCost, printPath, printSolutio, dijkstra lấy ý tưởng từ geeksforgeeks (<https://www.geeksforgeeks.org/printing-paths-dijkstras-shortest-path-algorithm/>) kết hợp với thuật toán đệ quy truyền số lần tìm đường đi ngắn nhất , phải lớn min của tổng chi phí chuyến bay lúc đầu là 0, sau sẽ theo min của chuyến bay ngắn nhất trước đó.

DocRequestFile: tương tự đọc như hàm DocFlightFile để lấy tên 2 thành phố và loại tìm kiếm, sau đó sử dụng hàm dijkstra để xử lý.

# File test:

Được làm từ số liệu thực tế ở trang WeGo( <https://www.wego.vn/lich-bay> ) chứa trong thư mục code.