CS162 - CSC10002 – Tasks for Solo Project

Class: 23APCS1-2, 23CLC03

# Week11

Tìm hiểu về các LLM models, tích hợp cho việc truy vấn gọi hàm bằng ngôn ngữ tự nhiên (tiếng Anh hoặc tiếng Việt).

# Week10

1. Cập nhật là code tìm đường đi ngắn nhất bằng Dijkstra ở Week07, bổ sung các thông số sau đây.
   1. Tổng thời gian tính metric: stop có độ quan trọng cao nhất.
   2. Tổng thời gian chạy cho all pairs.
   3. Thời gian trung bình, nhanh nhất, chậm nhất để chạy shortest path cho 1 đỉnh (ghi rõ là 1..n hay 1..1).
2. Cải tiến thuật toán tìm đường đi ngắn nhất, ghi nhận lại các thông số thời gian như trên. Chẳng hạn: Floyd–Warshall algorithm, Contraction Hierarchy….

# Week07

Dựa vào các file dữ liệu vars.json, stops.json và paths.json. SV xây dựng class và thực hiện các yêu cầu sau:

1. Xây dựng graph trong đó: đỉnh là stop, cạnh là các cung liền kề của tuyến xe bus, trọng số là thời gian (second). SV dựa vào giờ bắt đầu của tuyến xe, (lat, lng) để tính được thời gian “tương đối” giữa 2 đỉnh liền kề. SV nhớ đổi (lat, lng) sang (x, y) trước khi thực hiện. Với mỗi cạnh lưu cả trọng số thời gian (second) và trọng số khoảng cách (m).
2. Tự cài đặt giải thuật Dijkstra, tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh trong graph (all pairs). Lưu kết quả vào file.
3. Chọn (start\_stop, end\_stop), export dạng json đường đi ngắn nhất giữa 2 stop. Export kèm theo các path của các route tương ứng để đối chiếu đường đi có phù hợp hay ko?
4. Tính và hiển thị top k = 10 (default) stop có độ quan trọng cao nhất. Export dạng json có kèm theo id, số liệu, lat, lng, tên stop. Cách tính: cứ có 1 shortest path đi qua stop A thì count(stopA) tăng 1.

*Trong technical report, sv cần trình bày sơ đồ lớp, chi tiết cài đặt kèm theo thời gian chạy all pairs shortest path, chuẩn bị cho các cải tiến ở các tuần sau.*

# Week06

Cho file paths.json chứa thông tin lộ trình đi của các (route, var) ở TP.HCM.

Viết technical report, vẽ sơ đồ lớp (class diagram) và cài đặt (Python) cho các yêu cầu sau:

1. Sử dụng pyproj để chuyển đổi 1 tọa độ (lat, lng) về tọa độ (x, y). Xem thông tin về crs 3405.
2. Tìm hiểu trang web <https://geojson.io/>. Viết các file geojson để có thể vẽ 1 Point, 1 LineString… và load được trên trang web này.
3. Xây dựng class Path và PathQuery với các hàm tương tự như các tuần trước.
4. Tìm hiểu các hàm trong shapely.
5. Tìm hiểu rtree trong Python.
6. Tìm hiểu về các thư viện LLM để chọn hàm dựa vào query. Lưu ý, cẩn thận bị trừ nhiều tiền khi link tài khoản ngân hàng đến các thư viện (nếu có). Nên tìm và sử dụng các thư viện free.

# Week05

Cho file vars.json chứa thông tin các (route, var) ở TP.HCM

Route là một tuyến xe bus. VD: Tuyến xe bus số 6.

Var là một chiều của tuyến xe bus. VD: Tuyến xe bus số 6 có 2 chiều đi.

Mỗi (route, var) sẽ có nhiều chuyến xe chạy trong ngày.

Vẽ sơ đồ lớp (class diagram) và cài đặt (Python) cho các yêu cầu sau:

1. Xây dựng class RouteVar chứa thông tin một (route, var). SV: xem các thuộc tính trong file json.
   1. Xây dựng các constructors, các hàm get, set cho các thuộc tính của class RouteVar.
2. Xây dựng class RouteVarQuery để xử lí cho một file vars.json. Class là có thuộc tính là list các (route, var)
   1. Xây dựng hàm các hàm searchByABC() để truy vấn theo từng thuộc tính của (route, var), trả về list các (route, var) thỏa điều kiện.
   2. Xây dựng hàm outputAsCSV(list) để output 1 query list thành dạng CSV file.
   3. Xây dựng hàm outputAsJSON(list) để output 1 query list thành dạng JSON file. Mỗi object JSON là 1 dòng, tương tự như file vars.json.

Cho file stops.json chứa thông tin có thứ tự các bus stop của một (route, var).

Vẽ sơ đồ lớp (class diagram) và cài đặt (Python) cho các yêu cầu sau:

1. Xây dựng class Stop chứa thông tin một stop. SV: xem các thuộc tính trong file json.
   1. Xây dựng các constructors, các hàm get, set cho các thuộc tính của class Stop.
2. Xây dựng class StopQuery để xử lí cho một file stops.json. Class là có thuộc tính là list các stop.
   1. Xây dựng hàm các hàm searchByABC() để truy vấn theo từng thuộc tính của stop, trả về list các stop thỏa điều kiện.
   2. Xây dựng hàm outputAsCSV(list) để output 1 query list thành dạng CSV file.
   3. Xây dựng hàm outputAsJSON(list) để output 1 query list thành dạng JSON file. Mỗi object JSON là 1 dòng, tương tự như file stops.json.

# Week03

Viết report trình bày các nội dung sau.

1. Tạo GitHub PRIVATE repository
2. Clone Github về máy tính.
3. Tạo folder chứa source code.
4. Tạo file .gitignore.
5. Demo cách sử dụng git add, git commit, git push, git pull.
6. Tìm hiểu cách tạo project có nhiều file trong Python.
7. Tìm hiểu cách tạo classes, properties, methods.