ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

*

BÀI TẬP LỚN

MÔN HỌC: NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Đề tài: Phần mềm tìm kiếm đường đi giữa hai điểm trên bản đồ

Nhóm: 9

Mã lớp học: 157487

Giảng viên hướng dẫn: PGS.TS Trần Đình Khang

Danh sách sinh viên thực hiện:

STT	Họ và tên	MSSV	Email
1	Nguyễn Mạnh Thắng	20235426	thang.nm235426@sis.hust.edu.vn
2	Lưu Quang Nhật	20235392	nhat.lq235392@sis.hust.edu.vn
3	Đỗ Khắc Dũng	20235298	dung.dk235298@sis.hust.edu.vn
4	Vũ Đình Dũng	20235308	dung.vd235308@sis.hust.edu.vn
5	Nguyễn Ngọc Gia Bảo	20235268	bao.nng235268@sis.hust.edu.vn

Hà Nội – Ngày 18 tháng 5 năm 2025

Mục lục

Danh sách hình vẽ

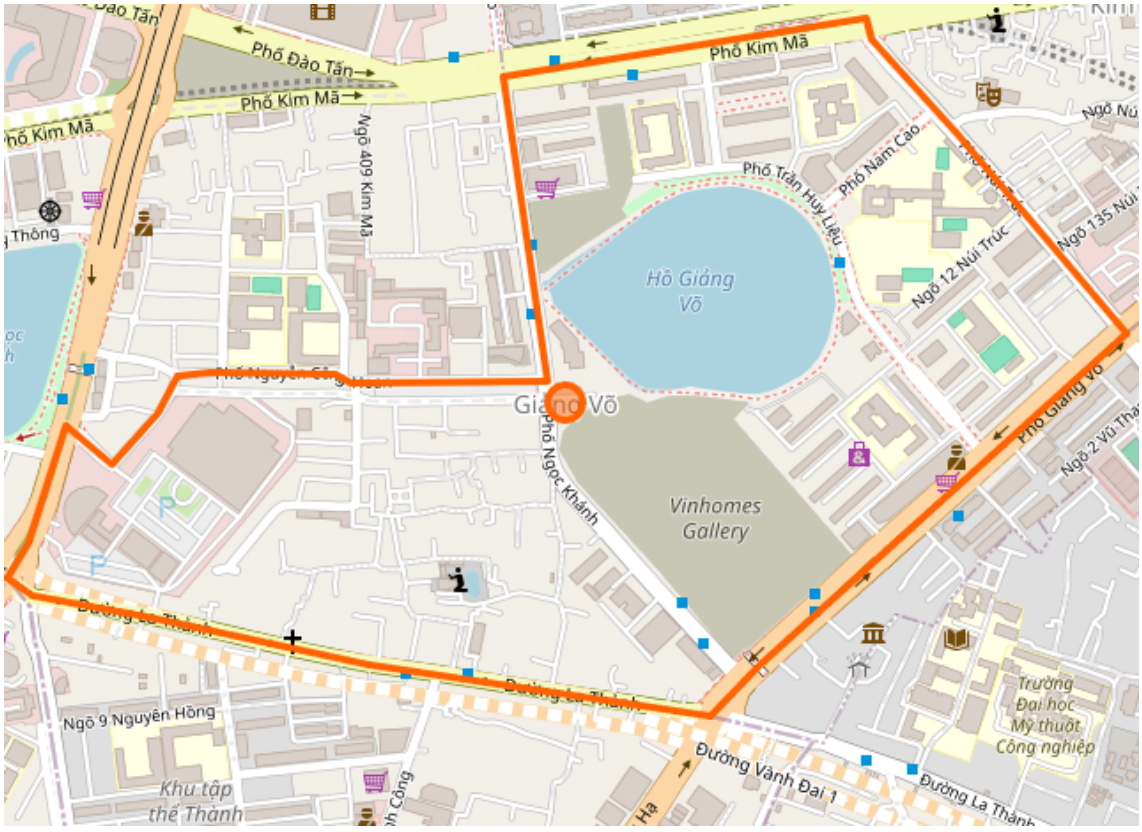
Chương 1

Mô tả bài toán

Nội dung của bài toán được miêu tả cụ thể như sau:

- Yêu cầu của bài toán: Tìm đường đi giữa hai điểm bất kỳ và biểu diễn chúng trên bản đồ.
- Không gian bài toán: Phường Giảng Võ, quận Ba Đình, Hà Nội.
- Các ràng buộc của bài toán là:
  - Không được đi một tuyến đường quá hai lần để tránh vòng lặp vô hạn.
  - Không được đi ngược đường một chiều.
- Các quyền của admin: cấm đường, cấm vùng, ...

Hình ảnh 1.1 dưới đây mô tả phạm vi của phường Giảng Võ trên bản đồ:



Hình 1.1: Phạm vi phường Giảng Võ

Chương 2
Biểu diễn bài toán

1 Mô tả đầu vào/đầu ra

Bài toán có input/output như sau:

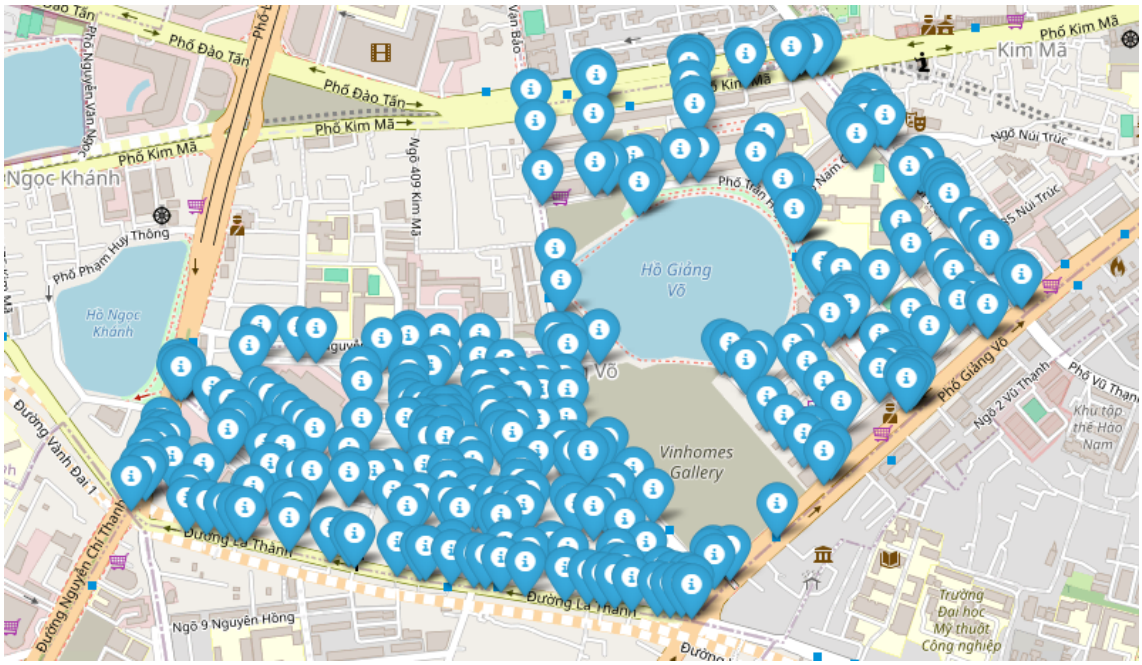
- **Input:** Admin xác nhận các đường, các vùng bị cấm. Người dùng click 2 điểm trên bản đồ.
- **Output:** Hiển thị đường đi tìm được giữa hai điểm người dùng chọn.

2 Biểu diễn các trạng thái

Các trạng thái của bài toán được biểu diễn là các vị trí trên bản đồ, thể hiện dưới dạng (*lat*, *long*) trong đó *lat*, *long* lần lượt thể hiện vĩ độ và kinh độ của vị trí đó trên bản đồ. Ví dụ:

- (21.028460, 105.821986) là vị trí THCS Giảng Võ.
- (21.024925, 105.816721) là vị trí của Đình Giảng Võ.

Các cặp tọa độ tương ứng với vị trí của điểm bắt đầu và điểm kết thúc sẽ lần lượt là trạng thái xuất phát và trạng thái đích của bài toán tìm đường đi. Trên hình 2.1, các trạng thái lần lượt được biểu diễn tương ứng với vị trí của chúng trên bản đồ phường Giảng Võ.

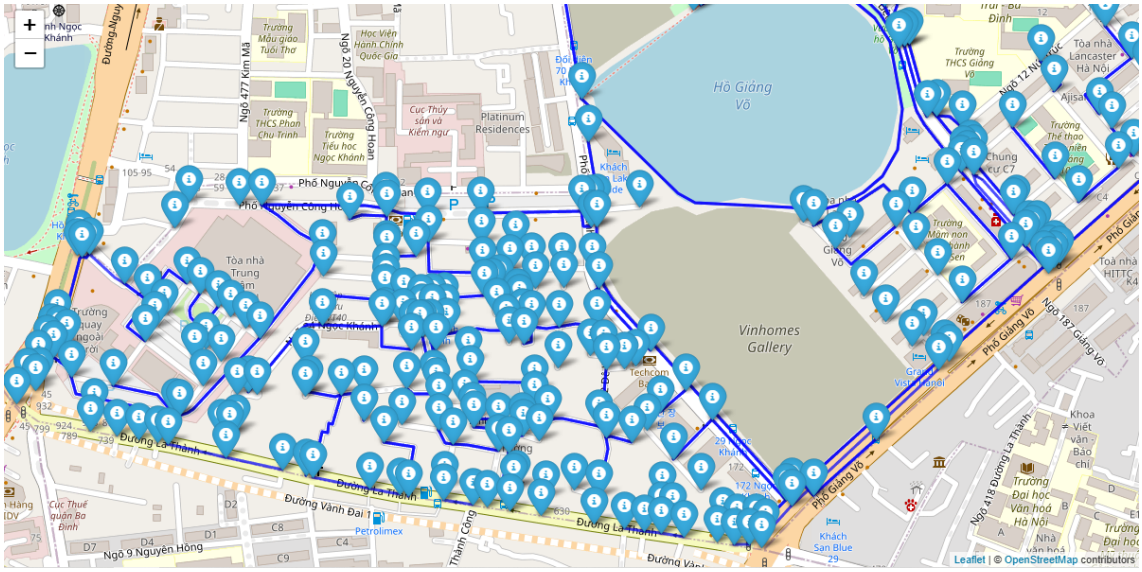


Hình 2.1: Trạng thái bài toán

3 Các hành động giữa các trạng thái

Tập các hành động là toàn bộ những cạnh trên đồ thị trạng thái, trong đó nối hai điểm có thể đi trực tiếp tới nhau thông qua con đường hợp lệ. Như vậy, đồ thị không gian trạng thái có thể được biểu diễn dưới dạng một đồ thị có hướng, trong đó một cạnh có hướng kết nối trạng thái *A* tới trạng thái *B* nếu từ *A* có thể đi trực

tiếp sang B. Đồ thị không gian trạng thái được biểu diễn ở hình 2.2, trong đó các đường màu xanh kết nối các trạng thái có thể chuyển tiếp.



Hình 2.2: Hành động trong bài toán

Chương 3
Phương pháp tìm kiếm

Trong chương này, chúng em sẽ sử dụng thuật toán A* để tìm kiếm đường đi giữa hai điểm trên bản đồ, với những tinh chỉnh bổ sung để phù hợp với các quyền giao cho admin: cấm đường, cấm vùng.

1 Phương pháp thực hiện

- **Bước 1:** Xử lý đầu vào đối với admin, xác nhận các tuyến đường bị cấm, xác nhận các vùng bị cấm.
  - **Bước 1.1:** Xác nhận các tuyến đường bị cấm. Admin 'click' vào một điểm gần tuyến đường muốn cấm, sau đó phần mềm sẽ đưa ra 5 tuyến đường gần điểm đó nhất. Admin sẽ tiến hành chọn tuyến đường phù hợp.
  - **Bước 1.2:** Xác nhận các vùng bị cấm. Admin 'click' vào một điểm trong vùng muốn cấm, sau đó phầm mềm sẽ yêu cầu nhập bán kính vùng đó. Admin sẽ tiến hành nhập bán kính phù hợp.
- **Bước 2:** Xử lý đầu vào đối với người dùng, biểu diễn trạng thái đầu và trạng thái đích.
  - **Bước 2.1:** Kiểm tra mọi điểm trong cơ sở dữ liệu (các trạng thái đã có dữ liệu), tính khoảng cách giữa điểm đó và tọa độ đầu vào.
  - **Bước 2.2:** Sau đó, xác định điểm gần nhất với mỗi tọa độ, và lấy đó làm trạng thái đầu và trạng thái đích.
- **Bước 3:** Sử dụng thuật toán tìm kiếm đường đi (*trình bày dưới đây*) để tìm đường đi từ trạng thái đầu đến trạng thái đích.

2 Thuật toán A*

Algorithm 1 Thuật toán A* (A-star)

function A-star(problem) **returns** a solution, or failure
$node \leftarrow$ a node with $State = problem.Initial\_State, Approx\_Cost = 0 + h(node)$
/*$h()$ is a heuristics function for each node, it returns an approximate cost to the target node*/
$frontier \leftarrow$ a priority queue ordered by $Approx\_Cost = Path\_Cost + h(node)$, with $node$ as the only element
$explored \leftarrow$ an empty set
is_segment_restricted(u, v) is a boolean function; if 1, segment(u, v) is restricted; else, segment(u, v) is not restricted
loop
    if Empty?$(frontier)$ **then return** failure
    end if
    $node \leftarrow$ Pop$(frontier)$ ▷ Chooses the lowest-cost node in frontier
    if $problem.Goal\_Test(node.State)$ **then return** Solution$(node)$
    end if
    Add $node.State$ to $explored$
    for all $action \in problem.Actions(node.State)$ **do**
        $child \leftarrow$ Child-Node$(problem, node, action)$
        if not is_segment_restricted$(node, child)$ **then**
            if $child.State \notin explored$ **and** $child.State \notin frontier$ **then**
                $frontier \leftarrow$ Insert$(child, frontier)$
            else if $child.State \in frontier$ **with higher** $Path\_Cost$ **then**
                Replace that $frontier$ node with $child$
            end if
        end if
    end for
end loop

Bộ khung của thuật toán A* (A-star) được mô tả trong Algorithm 1. Thuật toán này luôn tìm được lời giải tốt nhất nếu ta chọn được một heuristics chấp nhận được (h(n) luôn nhỏ hơn chi phí thực tế từ node n tới node đích h*(n)). Ở đây, chúng em sẽ chọn heuristics bằng 0, đảm bảo luôn tìm được đường đi tối ưu về độ dài. Nguyên tắc của thuật toán là: lặp lại quá trình mở rộng đỉnh có chi phí ước lượng tổng thể của đường đi qua đỉnh đó tới đích nhỏ nhất, cho đến khi mở rộng tới đích thì dừng.
Ở đây, chúng em cũng có một thay đổi bổ sung để phù hợp với các chức năng cung cấp: cấm đường, cấm vùng, đó là thêm một hàm kiểm tra đường cấm để đảm bảo rằng sẽ không bao giờ mở rộng tới các đỉnh là đỉnh của đường cấm.

Chương 4
Công cụ thực hiện

1 Giao diện

Phần giao diện được biểu diễn thông qua ứng dụng web, chạy trên thư viện Streamlit và sử dụng các thành phần bản đồ từ Folium để hiển thị bản đồ tương tác. Lớp bản đồ hiển thị mà ứng dụng đang sử dụng là OpenStreetMap (OSM). Giao diện sẽ hiển thị bản đồ, thông báo nếu click địa điểm nằm ngoài khoảng, thông báo lỗi, thông báo đã chọn thành công điểm đi/đến, thông báo quãng đường đã được chọn và độ dài quãng đường... Bản đồ sẽ hiển thị điểm đến, điểm đi, quãng đường đi dự kiến và quãng đường bị cấm.

2 Xử lý thao tác người dùng

Các thao tác của người dùng được xử lý chủ yếu thông qua các sự kiện tương tác trên giao diện Streamlit và bản đồ Folium:

- Chọn điểm trên bản đồ (Click để chọn điểm đi/đến hoặc cấm cấm đường).
- Bật/tắt chế độ cấm đường.
- Chọn các tuyến đường để cấm.
- Xác nhận hoặc huỷ cấm đoạn đường đã chọn.
- Xoá các đường đã cấm bằng click.
- Chọn lại điểm đi/đến.
- Bật/tắt chế độ cấm vùng.
- Điều chỉnh bán kính vùng cấm.
- ...

3 Xử lý logic

Các nội dung xử lý logic của ứng dụng (thuật toán tìm đường, lưu trữ dữ liệu về các nút, nút kề...) được lập trình bằng ngôn ngữ Python với sự hỗ trợ của thư viện OSMNX:

- Dữ liệu bản đồ được tải về từ OpenStreetMap và lưu dưới dạng file graphml, dữ liệu bao gồm toạ độ các điểm (nút) và các cạnh nối với các điểm kề nhau.
- Xử lý thuật toán tìm đường: Được xử lý nhờ thuật toán A* từ thư viên NetworkX(có sẵn trong OSMNX).
- Lưu trữ và xử lý các nút: dữ liệu các nút và các cạnh được lưu trong đối tượng đồ thị G, dùng để phục vụ cho việc hiển thị, tìm đường hoặc cấm đường.
- Cập nhật trạng thái bản đồ: khi người dùng cấm đường hoặc cấm vùng, ứng dụng sẽ cập nhật lại danh sách các cạnh bị cấm, khi tìm đường thì những đoạn đường đó sẽ bị loại khỏi quá trình tính toán.

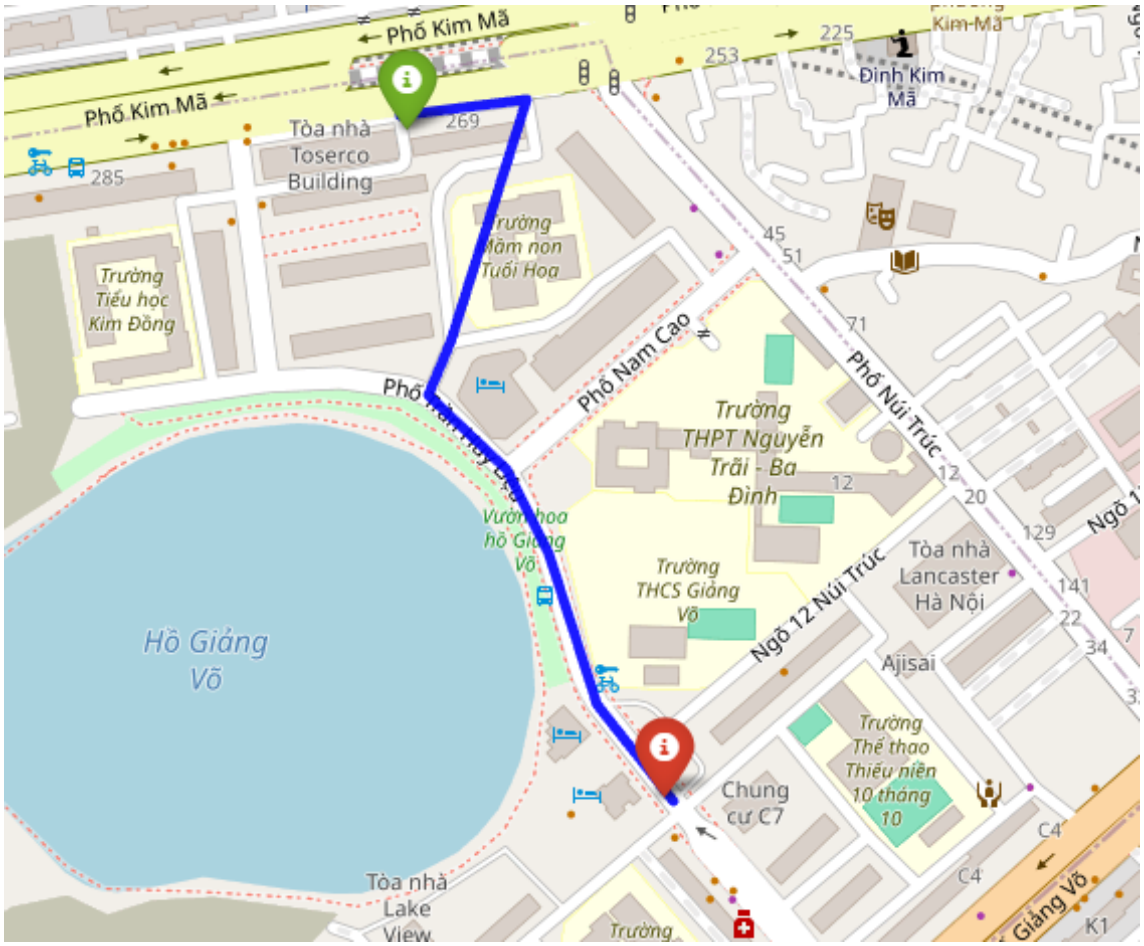
4 Mã nguồn

Link Github dự án: `https://github.com/ttt23/introai_project_map`

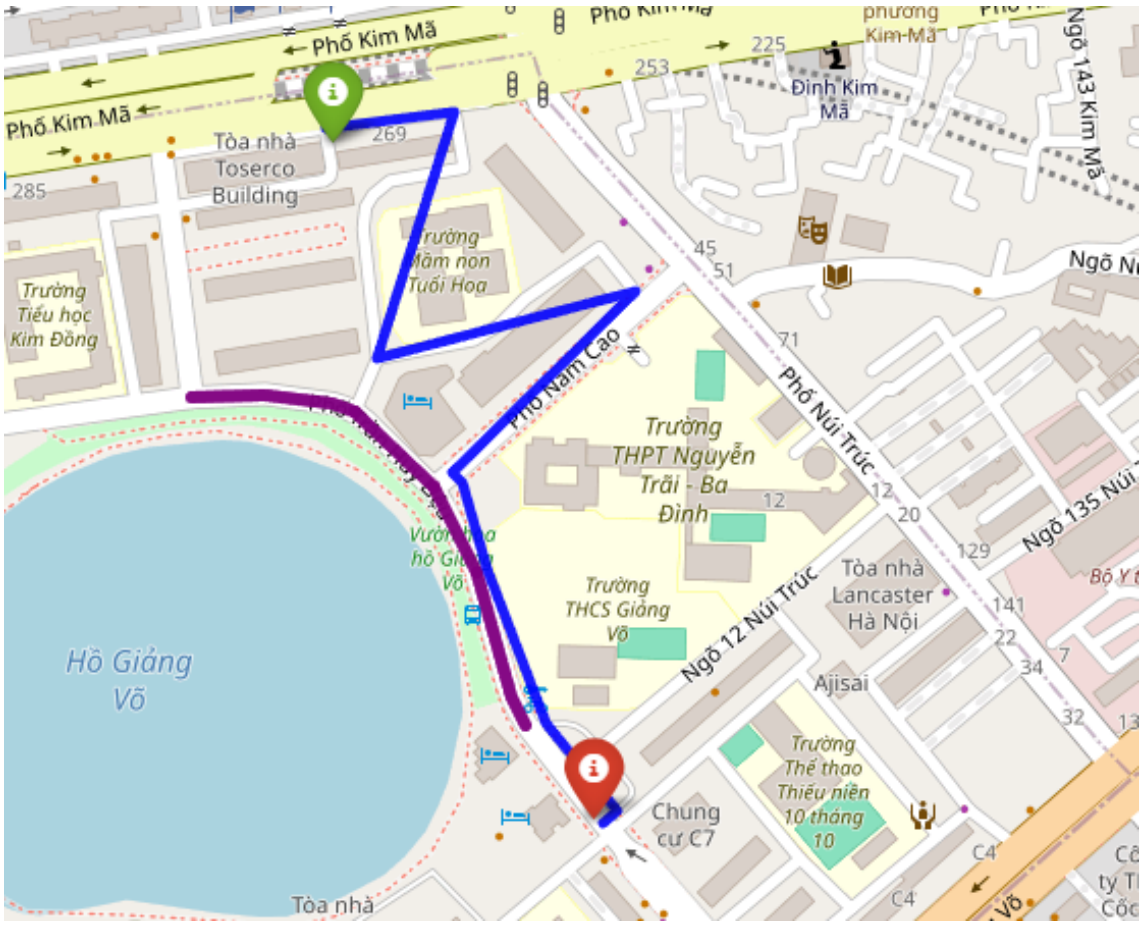
Chương 5

Kết quả

Trong chương này, chúng em sẽ trình bày kết quả dưới dạng hình ảnh, khi áp dụng thuật toán A*, kèm thêm với các chức năng: cấm đường, cấm vùng, để tìm đường đi giữa 2 nút trên bản đồ.
Hình ảnh 5.1, 5.2 sẽ trình bày đường đi tìm được trong lần lượt 2 tình huống: không có cấm đường và có cấm đường.
Lưu ý: đoạn đường màu xanh là đoạn đường tìm được, màu tím là đoạn đường bị cấm.

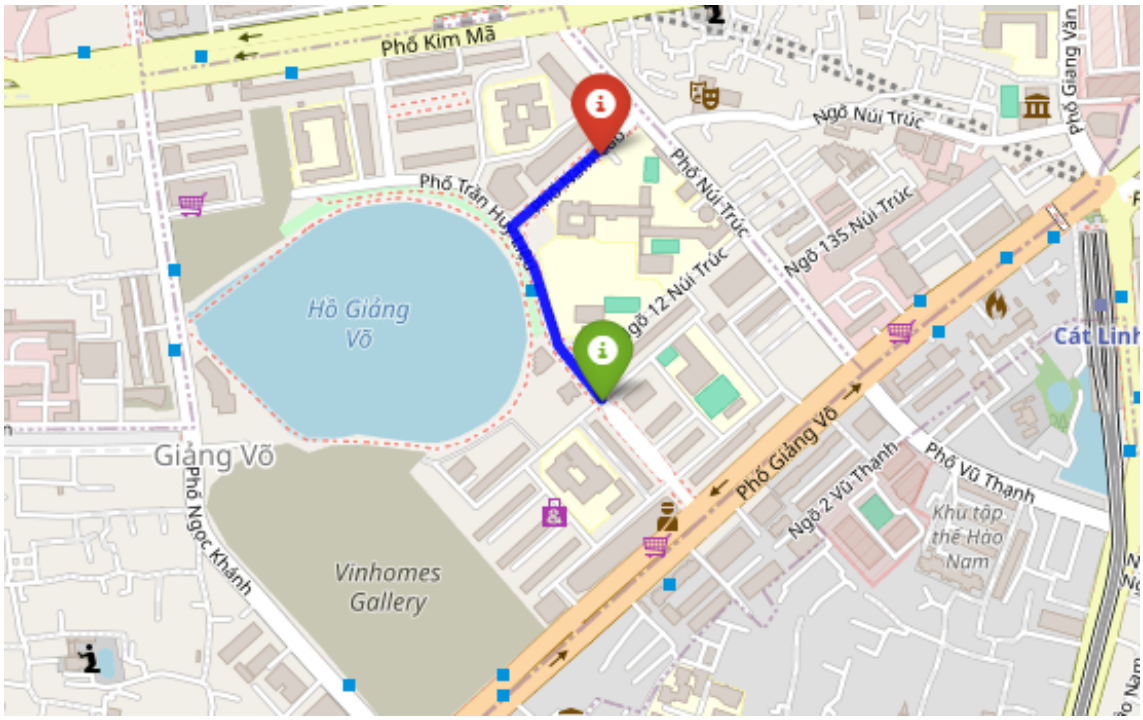


Hình 5.1: Đoạn đường đi tìm được trong TH chưa cấm đường

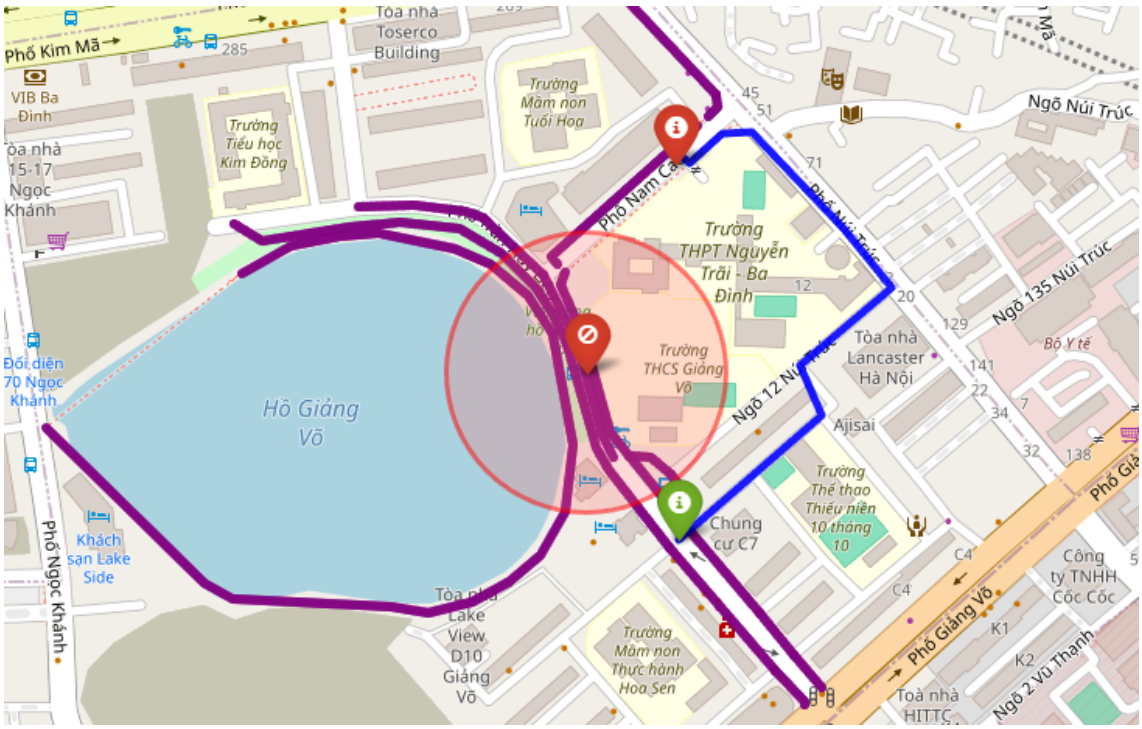


Hình 5.2: Đoạn đường đi tìm được trong TH cấm đường

Hình ảnh 5.3, 5.4 sẽ trình bày đường đi tìm được trong lần lượt 2 tình huống: không có cấm vùng và có cấm vùng.
Lưu ý: đoạn đường màu xanh là đoạn đường tìm được, màu tím là đoạn đường bị ảnh hưởng bởi vùng cấm không được phép di chuyển.



Hình 5.3: Đoạn đường đi tìm được trong TH chưa cấm vùng



Hình 5.4: Đoạn đường đi tìm được trong TH cấm vùng

Tài liệu tham khảo

[1] OpenStreetMap. https://www.openstreetmap.org.

[2] Streamlit documentation. https://docs.streamlit.io.

[3] Geoff Boeing. OSMNX documentation. https://osmnx.readthedocs.io.

[4] NetworkX. Networkx documentation. https://networkx.org/documentation/networkx-3.2.1/reference/index.html.

[5] Stuart J Russell and Peter Norvig. Artificial intelligence: a modern approach. Pearson, 2016.

[6] SOICT. Bài giảng soict. https://users.soict.hust.edu.vn/huonglt/AI/lecture%20notes.htm.

[7] Randy Zwitch. Streamlit-folium documentation. https://folium.streamlit.app.