

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



**ĐỒ ÁN MÔN HỌC**

---

**ĐỀ TÀI :**  
**HỆ THỐNG LƯU TRỮ VÀ THAO TÁC**  
**VỚI DỮ LIỆU CÁC CÂY ATM TẠI HÀ NỘI**

---

**Giảng viên hướng dẫn:**

TS. Vũ Tuyết Trinh

**Nhóm sinh viên thực hiện:**

Phùng Phú Cường - 20190084

Nguyễn Trần Hiếu Giang - 20191804

Nguyễn Đức Hiếu - 20192845

HÀ NỘI, 07/2023



## MỤC LỤC

<b>I. Giới thiệu đề tài</b>	2
<b>II. Phạm vi dữ liệu và chức năng của hệ thống</b>	2
1. Phạm vi dữ liệu	2
2. Cấu trúc dữ liệu	2
a. Dữ liệu cây ATM tại Hà Nội (atm_table)	2
b. Dữ liệu đường đi tới các cây ATM gần nhất (route_results)	2
3. Chức năng cơ bản của hệ thống	3
4. Hướng phát triển thêm các chức năng	3
<b>III. Các nghiên cứu liên quan</b>	3
1. PostgreSQL và PostGIS	3
2. QGIS	3
3. OpenStreetMap	3
4. Thuật toán tìm đường (Thuật toán Dijkstra)	3
<b>IV. Triển khai bài toán</b>	4
1. Thu thập dữ liệu	4
a. Dữ liệu cây ATM tại Hà Nội	4
b. Mô phỏng hàng chờ	4
2. Trình tự thực hiện	4
3. Chức năng tìm kiếm	5
a. Xác định những cây ATM gần nhất	5
b. Xác định đường đi ngắn nhất	6
c. Xác định đường đi tới cây ATM thỏa mãn	7
<b>V. Thực nghiệm</b>	8
1. Chuẩn bị dữ liệu các cây ATM tại Hà Nội	8
2. Kết quả tìm kiếm	9
a. Đường đi tới cây ATM có quãng đường ngắn nhất	9
b. Đường đi tới cây ATM có thời gian chờ ngắn nhất	9
<b>VI. Kết luận đánh giá</b>	10

## I. Giới thiệu đề tài

Với những người thường xuyên phải giao dịch qua hệ thống cây ATM của ngân hàng, đặc biệt là bằng đường bộ, các ứng dụng chỉ đường, tìm kiếm cây ATM sẽ là công cụ đắc lực giúp tìm đường ngắn nhất, tránh các điểm ùn tắc và lái xe an toàn, giúp công việc thanh toán, giao dịch được nhanh chóng và tiện lợi hơn.

Bài toán lưu trữ và thao tác với dữ liệu các cây ATM trên địa bàn Hà Nội sẽ giúp giải quyết các yêu cầu cơ bản như tìm tất cả các cây ATM trên địa bàn Hà Nội, tìm các cây ATM theo tên ngân hàng, tìm cây ATM gần nhất,...

## II. Phạm vi dữ liệu và chức năng của hệ thống

### 1. Phạm vi dữ liệu

- Thu thập dữ liệu về vị trí địa lý của cây ATM trên địa bàn thành phố Hà Nội.
- Thu thập dữ liệu về thành phố Hà Nội trên OpenStreetMap và import vào PostgreSQL.

### 2. Cấu trúc dữ liệu

#### a. Dữ liệu cây ATM tại Hà Nội (atm\_table)

- Số thứ tự của cây ATM
- Tên ngân hàng / chi nhánh của cây ATM
- Vị trí địa lý của cây ATM (loại dữ liệu: Point)
- Hàng chờ của cây ATM

#### b. Dữ liệu đường đi tới các cây ATM gần nhất (route\_results)

- Số thứ tự của cây ATM được xét
- Quãng đường từ điểm truy vấn tới cây ATM
- Hàng chờ của cây ATM
- Thời gian phải chờ sau khi di chuyển tới cây ATM
- Dữ liệu địa lý của đường đi (loại dữ liệu: MultiLine)

### 3. Chức năng cơ bản của hệ thống

- Tìm tất cả các cây ATM trên địa bàn Hà Nội.
- Tìm các cây ATM có đường đi ngắn nhất.
- Tìm cây ATM có đường gần nhất và hàng chờ vắng nhất.
- Tìm cây ATM có đường gần nhất và thời gian đợi ngắn nhất.

### 4. Hướng phát triển thêm các chức năng

Mô phỏng số lượng người rút tiền đang chờ ở các cây ATM khi thực hiện truy vấn, từ đó tính tổng thời gian đi và chờ đến khi được rút tiền để tìm cây ATM tiết kiệm thời gian nhất, tránh tình trạng ùn tắc và chờ đợi.

## III. Các nghiên cứu liên quan

### 1. PostgreSQL và PostGIS

- PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng để lưu thông tin về các ATM và dữ liệu bản đồ Hà Nội.
- PostGIS là 1 phần mở rộng của hệ quản trị CSDL PostgreSQL để quản lý không gian.

### 2. QGIS

- QGIS : phần mềm miễn phí về hệ thống thông tin địa lý. Tính năng chính của QGIS là thao tác trên các lớp bản đồ có dạng véc-tơ. QGIS được sử dụng để thể hiện các kết quả truy vấn của bài toán tìm đường.

### 3. OpenStreetMap

- OpenStreetMap (OSM) : dịch vụ bản đồ thế giới trực tuyến miễn phí. OSM được sử dụng trong quá trình khai thác và chuẩn bị dữ liệu cần thiết cho bài toán.

### 4. Thuật toán tìm đường (Thuật toán Dijkstra)

Thuật toán Dijkstra là một trong những thuật toán cổ điển để giải quyết bài toán tìm đường đi ngắn nhất từ một điểm cho trước tới tất cả các điểm còn lại trong đồ thị có trọng số.

## IV. Triển khai bài toán

### 1. Thu thập dữ liệu

#### a. Dữ liệu cây ATM tại Hà Nội

- Thu thập dữ liệu về tên, vị trí địa lý của toàn bộ cây ATM trong thành phố Hà Nội từ OSM

- Khởi tạo cột chứa giá trị hàng chờ của từng cây ATM

- Khởi tạo index trên số thứ tự và vị trí địa lý của cây ATM

#### b. Mô phỏng hàng chờ

- Dữ liệu hàng chờ được mô phỏng dựa vào thời điểm truy vấn, dựa vào các yếu tố như thời điểm trong ngày (giờ hành chính / ngoài hành chính), ngày trong tuần (ngày làm việc / cuối tuần), ngày trả lương và từ đó sinh ra độ bận rộn của cây ATM,

- Với độ bận rộn từ 0 đến 3, cây ATM sẽ có hàng chờ như sau :

- Với độ bận rộn 0 : Cây ATM sẽ có từ 0 đến 3 người

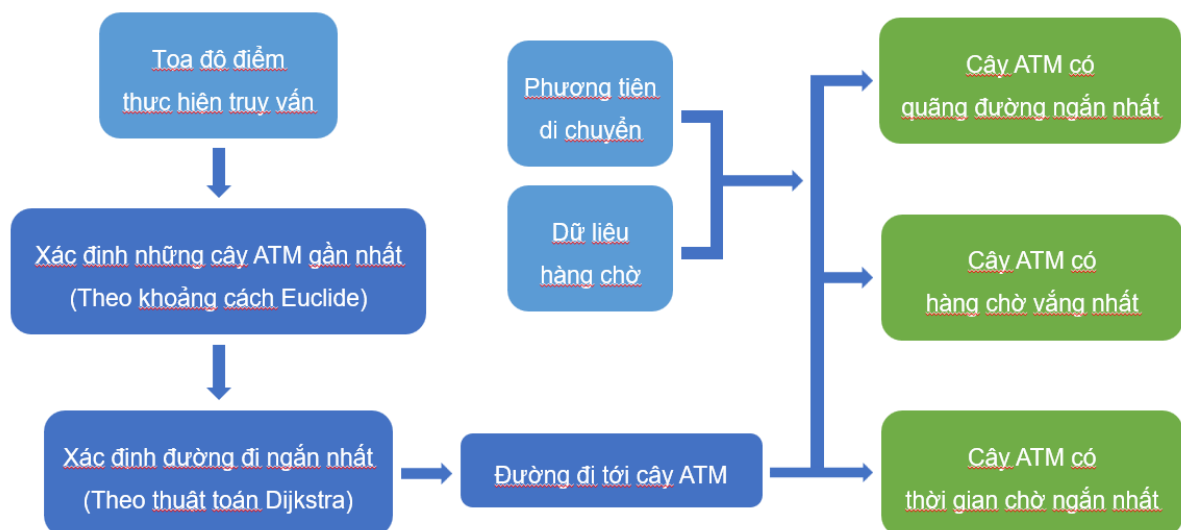
- Với độ bận rộn 1 : Cây ATM sẽ có từ 0 đến 4 người

- Với độ bận rộn 2 : Cây ATM sẽ có từ 0 đến 5 người

- Với độ bận rộn 3 : Cây ATM sẽ có từ 1 đến 5 người

- Thời gian phải đợi tại cây ATM sẽ được tính thông qua việc giả định mỗi người tại cây ATM sẽ mất từ 3 đến 5 phút để hoàn thành thao tác.

### 2. Trình tự thực hiện





### 3. Chức năng tìm kiếm

#### a. Xác định những cây ATM gần nhất

- Đưa vào tham số tọa độ điểm truy vấn
- Thực hiện truy vấn trên bảng dữ liệu cây ATM
- Sắp xếp và lấy ra những cây ATM gần nhất dựa trên khoảng cách Euclide từ điểm truy vấn tới cây ATM (mặc định là 10)
- Truy vấn 10 cây ATM gần nhất dựa trên khoảng cách Euclide :

Trường hợp 1: Không có index trong bảng atm\_table

	id [PK] integer	way geometry	queue smallint
1	345	0101000020110F0000068D6D70F2776641CBC66CF25E4542...	0
2	340	0101000020110F0000192D0680FA7766413D94EB5489454241	2
3	341	0101000020110F0000192D0680FA7766413D94EB5489454241	1
4	344	0101000020110F0000192D0680FA7766413D94EB5489454241	2
5	343	0101000020110F0000192D0680FA7766413D94EB5489454241	2
6	342	0101000020110F0000192D0680FA7766413D94EB5489454241	3
7	339	0101000020110F0000192D0680FA7766413D94EB5489454241	0
8	338	0101000020110F000010F0472CF477664159B3513E8D454241	2
Total rows: 10 of 10		Query complete 00:00:00.274	

Trường hợp 2: Index ở trên cột id của bảng atm\_table

	id [PK] integer	way geometry	queue smallint
1	345	0101000020110F0000068D6D70F2776641CBC66CF25E4542...	2
2	344	0101000020110F0000192D0680FA7766413D94EB5489454241	1
3	340	0101000020110F0000192D0680FA7766413D94EB5489454241	0
4	341	0101000020110F0000192D0680FA7766413D94EB5489454241	1
5	339	0101000020110F0000192D0680FA7766413D94EB5489454241	2
6	343	0101000020110F0000192D0680FA7766413D94EB5489454241	1
7	342	0101000020110F0000192D0680FA7766413D94EB5489454241	0
8	338	0101000020110F000010F0472CF477664159B3513E8D454241	3
Total rows: 10 of 10		Query complete 00:00:00.265	



### Trường hợp 3: Index ở trên cột id và cột way của bảng atm\_table

	id [PK] integer	way geometry	queue smallint
1	345	0101000020110F0000068D6D70F2776641CBC66CF25E4542...	2
2	344	0101000020110F0000192D0680FA7766413D94EB5489454241	1
3	340	0101000020110F0000192D0680FA7766413D94EB5489454241	0
4	341	0101000020110F0000192D0680FA7766413D94EB5489454241	1
5	339	0101000020110F0000192D0680FA7766413D94EB5489454241	2
6	343	0101000020110F0000192D0680FA7766413D94EB5489454241	1
7	342	0101000020110F0000192D0680FA7766413D94EB5489454241	0
8	338	0101000020110F000010F0472CF477664159B3513E8D454241	3
Total rows: 10 of 10		Query complete 00:00:00.263	



#### b. Xác định đường đi ngắn nhất

- Đưa vào tham số tọa độ điểm truy vấn và vị trí cây ATM được xét
- Xác định cạnh gần nhất với 2 điểm trên, lấy gốc của mỗi cạnh làm đỉnh xuất phát và đỉnh kết thúc
- Áp dụng thuật toán Dijkstra tìm đường đi ngắn nhất
- Bổ sung đường nối điểm truy vấn – đỉnh xuất phát và đỉnh kết thúc – cây ATM để có đường đi đầy đủ
- Trường hợp không tìm được đường đi do điểm truy vấn và cây ATM quá gần nhau, thay thế bằng đường nối từ điểm truy vấn tới cây ATM
- Tìm đường đi gần nhất từ điểm truy vấn tới cây ATM chỉ định

#### Trường hợp 1: Không có index trong bảng atm\_table



	distance_meter double precision	geom_path geometry
1	636.8471288521575	0105000020E6100000070000
Total rows: 1 of 1		
Query complete 00:00:00.568		

## Trường hợp 2 : Index trên cột id của bảng atm\_table

	distance_meter double precision		 geom_path geometry
1	636.8471288521575		0105000020E61000000700

Total rows: 1 of 1    Query complete 00:00:00.526

## Trường hợp 3: Index trên cột id và cột way của bảng atm\_table

	distance_meter double precision		 geom_path geometry
1	636.8471288521575		0105000020E61000000700

Total rows: 1 of 1    Query complete 00:00:00.504

### c. Xác định đường đi tới cây ATM thỏa mãn

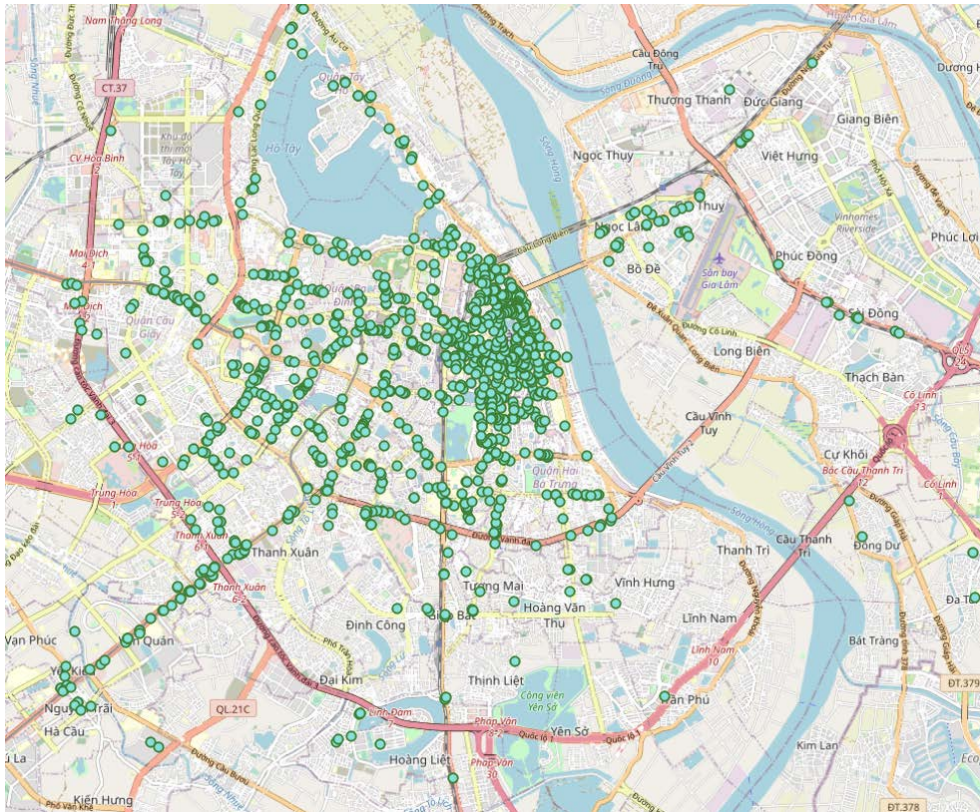
- Đưa vào tham số phương tiện di chuyển và dữ liệu hàng chờ
- Xác định cây ATM có quãng đường ngắn nhất / hàng chờ vắng nhất / thời gian chờ ngắn nhất :
  - Cây ATM có quãng đường ngắn nhất : Sắp xếp theo quãng đường cần di chuyển
  - Cây ATM có hàng chờ vắng nhất: Sắp xếp theo hàng chờ hiện tại của cây ATM
  - Cây ATM có thời gian chờ ngắn nhất: Sắp xếp theo thời gian phải chờ sau khi di chuyển tới cây ATM



- Đưa ra kết quả truy vấn: Đường đi, quãng đường cần di chuyển, thời gian di chuyển, hàng chờ tại thời điểm truy vấn, thời gian phải chờ sau khi di chuyển

## V. Thực nghiệm

### 1. Chuẩn bị dữ liệu các cây ATM tại Hà Nội

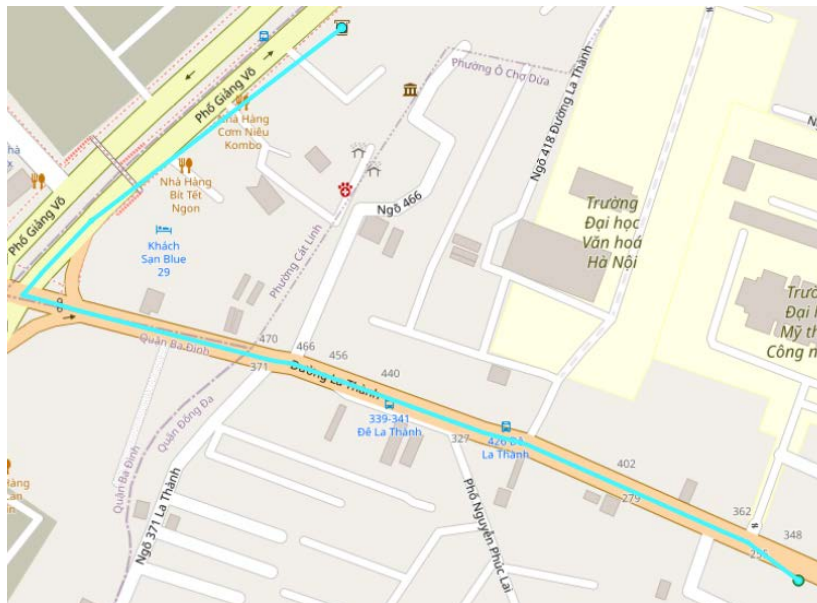


- Sau khi có dữ liệu được trích xuất từ file Vietnam.osm, lọc và lưu trữ dữ liệu liên quan đến các cây ATM tại Hà Nội
- Có tổng cộng 1259 điểm là các cây ATM có tên ngân hàng
- Thời gian truy vấn là 118ms

## 2. Kết quả tìm kiếm

### a. Đường đi tới cây ATM có quãng đường ngắn nhất

- Lấy tọa độ truy vấn 21.0221037,105.8236547 và phương tiện là xe máy



Mean of Transport: Motor (30 km/h)  
Distance: 632.0 m  
Travel Time: 1 minutes  
Current Queue: 2 people  
Estimated Wait Time: 6.0 minutes

### b. Đường đi tới cây ATM có thời gian chờ ngắn nhất

- Lấy tọa độ truy vấn 21.0221037,105.8236547 và phương tiện là xe máy



Mean of Transport: Motor (30 km/h)  
Distance: 729.0 m  
Travel Time: 1 minutes  
Current Queue: 0 people  
Estimated Wait Time: 0.0 minutes

## VI. Kết luận đánh giá

- Kết luận : Kết quả truy vấn đạt yêu cầu cơ bản
- Đánh giá :
  - Đường đi tìm được đáp ứng với yêu cầu đặt ra của bài toán.
  - Đường đi và khoảng cách được thể hiện chưa chính xác 100% do hạn chế về mặt thuật toán.
  - Cần xét thêm mật độ giao thông để có kết quả chính xác hơn khi tính toán thời gian di chuyển, thời gian phải chờ tại mỗi cây ATM.