**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

A red and white logo

Description automatically generated

**BÀI TẬP LỚN**

**MÔN HỌC:** NHẬP MÔN TRÍ TUỆ NHÂN TẠO

**MÃ MÔN HỌC:** IT3160

**MÃ LỚP HỌC:** 144916

**GIÁO VIÊN HƯỚNG DẪN:** PGS.TS. Trần Đình Khang

**NHÓM 23 – DANH SÁCH SINH VIÊN**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Phan Đức Thịnh | 20204693 |
| 2 | Lê Xuân Giao | 20210290 |
| 3 | Phạm Bùi Hải | 20215362 |
| 4 | Bùi Anh Minh | 20215422 |
| 5 | Phạm Đình Tú | 20210888 |

*Hà Nội – 12/2023*

**Lời cảm ơn:**

Em xin chân thành cảm ơn PGS.Ts Trần Đình Khang đã cung cấp tài liệu cũng như giảng dạy môn học giúp em hiểu biết và có kiến thức về trí tuệ nhân tạo từ đó tạo ra project này. Em cũng xin cảm ơn trường Công Nghệ Thông Tin và Truyền Thông đã đưa môn học vào giảng dạy để giúp em có cơ hội được tiếp cận với môn học đang trở thành xu thế của thời đại mới.   
Do kiến thức còn hạn chế và kinh nghiệm chưa có nhiều, nên bài báo cáo khó tránh khỏi thiếu sót. Em rất mong nhận được những nhận xét từ thầy để bài báo cáo được hoàn thiện hơn.

**MỤC LỤC**

[**I. Đóng góp của thành viên** 3](#_Toc153320247)

[**II.** **Giới thiệu** 4](#_Toc153320248)

[1. Bối cảnh 4](#_Toc153320249)

[2. Mô tả bài toán 4](#_Toc153320250)

[3.Định nghĩa bài toán 4](#_Toc153320251)

[**III. Lý thuyết** 6](#_Toc153320252)

[1. Thuật toán A\* 6](#_Toc153320253)

[2. Lý do chọn thuật toán: 6](#_Toc153320254)

[**IV. Triển khai** 6](#_Toc153320255)

[1. Front-end: Django, html, css, js 6](#_Toc153320256)

[2. Lõi thuật toán: python 7](#_Toc153320257)

[3. Bản đồ: google map api 7](#_Toc153320258)

[4. File tọa độ xml: OpenStreetMap 7](#_Toc153320259)

[5. Hàm tính khoảng cách giữa 2 điểm: Haversine formula 7](#_Toc153320260)

[**V Kết quả đạt được** 7](#_Toc153320261)

[1. Link github 7](#_Toc153320262)

[2. Giao diện cho phép người dùng chọn điểm bắt đầu và điểm đích: 8](#_Toc153320263)

[3. Giao diện hiển thị đường đi ngắn nhất 9](#_Toc153320264)

[4. Bản đồ cho phép xem full màn hình, phóng to, thu nhỏ tạo sự tiện lời cho người dùng 9](#_Toc153320265)

[**VI Hướng phát triển** 11](#_Toc153320266)

[**VII Tài liệu tham khảo** 11](#_Toc153320267)

# **I. Đóng góp của thành viên**

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | Đóng góp |
| 1 | Phan Đức Thịnh | Code: 100%, báo cáo 50%, slide: 100% |
| 2 | Lê Xuân Giao | 0% |
| 3 | Phạm Bùi Hải | 0% |
| 4 | Bùi Anh Minh | Báo Cáo 50% |
| 5 | Phạm Đình Tú | 0% |

# **II.** **Giới thiệu**

## 1. Bối cảnh

Bài toán tìm đường đi trên đồ thị là một trong những bài toán đa dạng, có tính ứng dụng cao và áp dụng vào rất nhiều trong thực tế: ví dụ các app bản đồ, app giao hàng, thương mại điện tử,…

Một trong những ứng dụng đó là tìm kiếm đường đi giữa hai địa điểm cụ thể. Để giải quyết vấn đề trên, các ứng dụng bản đồ như Google Maps, Bing Maps,… sử dụng một số thuật toán (Dijkstra, A\*,…) để tính toán đường đi tối ưu nhất giữa hai địa điểm bất kỳ về mặt đường đi . Các ứng dụng bản đồ mang một vai trò quan trọng khi nó giúp người dùng tính toán chi phí hay thời gian đi lại một cách hiệu quả, đặc biệt là những người thường xuyên đi du lịch.

## 2. Mô tả bài toán

Cho bản đồ các tuyến đường và địa điểm của một nơi trong Hà Nội, trong bài toán này chúng em chọn địa điểm quanh khu vực Trường Chinh Hà Nội.

Cùng với các hàm ước lượng h(i, j) là khoảng cách từ điểm i đến điểm j.

Một du khách nước ngoài vừa mới đặt chân tới một nơi gầ Trường Chinh Hà Nội được ít hôm. Lúc này anh ấy đang ở một địa điểm X nào đó, để thuận tiện cho việc trải nghiệm và khám phá, anh muốn di chuyển tới địa điểm Y sao cho khoảng cách di chuyển giữa hai địa điểm là nhỏ nhất. Thực hiện tìm kiếm đường đi theo giải thuật A\* từ địa điểm X tới địa điểm Y theo yêu cầu trên.

## 3.Định nghĩa bài toán

a. Xác định PEAS

* Đánh giá hiệu quả hoạt động (P): cực tiểu hóa độ dài di chuyển giữa hai địa điểm bất kỳ
* Môi trường xung quanh (E): các nút giao, tuyến đường, địa điểm
* Các bộ phận hành động (A): hiển thị trên màn hình bản đồ, đường đi ngắn nhất,…
* Các bộ phận cảm biến (S): chuột, màn hình.

b. Xác định các thuộc tính của tác tử (agent)

* Các trạng thái: (đích, không phải đích)
* Các hành động: di chuyển giữa các tuyến đường

c. Phát biểu bài toán

* Mục tiêu: cần phải có mặt ở địa điểm Y
* Trạng thái ban đầu: “đang ở địa điểm X”
* Các hành động: xác định bởi hàm chuyển trạng thái S(i) = tập các cặp <i → j, j>
* Kiểm tra mục tiêu: trạng thái hiện thời = “đang ở địa điểm Y”
* Tìm kiếm giải pháp: tuyến đường ngắn nhất cần đi
* Giải pháp: đường đi ngắn nhất giữa điểm bắt đầu và đích

# **III. Lý thuyết**

## 1. Thuật toán A\*

* Thuật toán A\* là thuật toán tìm đường đi dựa trên so sánh hàm tổng f = g + h. Trong đó f là hàm để so sánh chi phí, g là chi phí thực tế, h là 1 hàm ước lường (heuristic).
* Đối với bài toán tìm đường đi ngắn nhất:
* g là khoảng cách đã đi thực tế từ điểm bắt đầu đến điểm hiện tại
* h là khoảng cách ước lượng theo đường chim bay từ điểm hiện tại đến điểm đích.

## 2. Lý do chọn thuật toán:

* Đối với bài toán tìm đường đi trong thực tế, có cực kỳ nhiều điểm, đường ta cần đánh dấu và lưu lại, nên nếu dùng các thuật toán khác như BFS, DFS thì số lượng phép tính và thời gian thực hiện sẽ rất lâu và tốn chi phí, bên cạnh đó thuật toán greedy best first search có thể gây ra tình trạng bị vòng lặp vô hạn. Chính vì thế ta sử dụng thuật toán A\* giúp tìm kiếm 1 cách nhanh chóng hơn và tối ưu hơn dựa trên hàm độ ưu tiên được tính bằng khoảng cách thực tế + hàm heuristic ước lượng khoảng cách. Và với số node và way là hữu hạn thuật toán A\* là thuật toán hoàn chỉnh và sẽ luôn tìm được lời giải nếu bài toán có.

# **IV. Triển khai**

Để triển khai bài toán chúng em dùng các công nghệ:

## 1. Front-end: Django, html, css, js

Trong đó:  
+ Django là 1 framework của python để xây dựng các web một cách nhanh chóng.

+ Html, css là những công nghệ để xây dựng giao diện, viết Style cho giao diện

+ Js là ngôn ngữ để viết script cho giao diện

## 2. Lõi thuật toán: python

python là một ngôn ngữ lập trình ứng dụng cao đặc biệt trong Ai với cộng đồng lớn, dễ sử dụng.

## 3. Bản đồ: google map api

Google map api là thư viện để hiển thị bản đồ ra giao diện

## 4. File tọa độ xml: OpenStreetMap

OpenStreetMap là 1 nền tảng mã nguồn mở cho phép lấy tọa độ dưới dạng file xml

## 5. Hàm tính khoảng cách giữa 2 điểm: Haversine formula

Đây là 1 hàm tính khoảng cách giữa 2 điểm dựa vào tọa độ của 2 điểm trên trái đất.

- Cụ thể: Từ file xml tọa độ của các điểm và đường trong OpenStreetMap, ta dùng Etree của python để convert nó thành các node và các edge và từ đó xây dựng nên 1 đồ thị cạnh kề giữa các điểm(gồm các điểm, các hàng xóm của nó và khoảng cách của mỗi điểm đến hàng xóm). khi người dùng bật trang web lên, màn hình hiển thị 1 bản đồ của khu vực quanh Trường Chinh Hà Nội được lấy từ google api, khi người dùng nhấn chọn 2 điểm gồm đích đến và điểm bắt đầu, hệ thống sẽ lấy vị trí tọa độ(kinh độ, vĩ độ) của 2 điểm gần nhất đã được đánh dấu và lưu, sau đó dùng thuật toán A\* với heuristic là khoảng cách giữa điểm hiện tại với điểm đích. Từ đó tìm được 1 path gồm các điểm cần đi qua để đi đến đích ngắn nhất, sau đó hiển thị đường đi lên màn hình kết quả.

# **V Kết quả đạt được**

## 1. Link github

<https://github.com/WhiteCloudHao/project_ai_find_path>

## 2. Giao diện cho phép người dùng chọn điểm bắt đầu và điểm đích:

A map with text and images

Description automatically generated  
H IV.1 Màn hình giao diện bản đồ cho phép người dùng chọn

A screenshot of a map

Description automatically generated

H IV.2: Khi người dùng chọn, sẽ hiển thị tọa độ các điểm bắt đầu và điểm đích

## 3. Giao diện hiển thị đường đi ngắn nhất

A map with red points and white text

Description automatically generated

H IV.3 giao diện hiển thị đường đi ngắn nhất giữa 2 điểm

## 4. Bản đồ cho phép xem full màn hình, phóng to, thu nhỏ tạo sự tiện lời cho người dùng

A map with red and black points

Description automatically generated  
H IV.4 Bản đồ cho phép xem full màn hình

A screenshot of a map

Description automatically generated

H IV.5 Bản đồ cho phép phóng to, thu nhỏ

# **VI Hướng phát triển**

Hướng phát triển tiếp theo của dự án là sẽ tiếp tục phát triển các chức năng đã có, làm cho nó tối ưu, thân thiện hơn với người dùng. Sau giai đoạn này, dự án sẽ tiếp tục nâng cấp thêm các chức năng như  
 - tìm đường đi ít thời gian nhất, báo cáo tình trạng giao thông, báo cáo chi tiết về độ dài, số km đã đi,…

* Chia sẻ vị trí và tìm đường đi realTime giữa 2 người với nhau.
* Xây dựng các thư viện, nền tảng cho phép tạo các đường đi cá nhân,…

# **VII Tài liệu tham khảo**

[1] Bài giảng của PGS.TS Trần Đình Khang

[2] Thuật toán A\* : [https://en.wikipedia.org/wiki/A\*\_search\_algorithm](https://en.wikipedia.org/wiki/A*_search_algorithm)

[3] Framework Django: <https://www.djangoproject.com/>

[4] Open street map: <https://www.openstreetmap.org/#map=16/20.9960/105.8347>

[5] Công thức Haversine formula: <https://en.wikipedia.org/wiki/Haversine_formula>