|  |  |
| --- | --- |
| Trường Đại học Khoa học Tự Nhiên  **ĐHQG.TPHCM** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập - Tự do - Hạnh phúc**  *TPHCM,ngày 24 tháng 11 năm 2024* |

**BÁO CÁO THỰC HÀNH – NHẬP MÔN TRÍ TUỆ NHÂN TẠO – LẦN 4**

**Họ và tên: Nguyễn Thanh Kiên.**

**MSSV: 22110092.**

**Nhắc lại kiến thức**

1. Thuật toán BFS:

A screenshot of a computer

Description automatically generated

1. Thuật toán DFS:

A white paper with black text

Description automatically generated

1. Thuật toán UCS:

A black and white text

Description automatically generated

1. Thuật toán GBFS:

A screenshot of a computer code

Description automatically generated

1. Thuật toán A\*

A black and white page with text

Description automatically generated

**Nhận xét tổng quan:**

1. **Bài toán:**

Bài toán được định nghĩa như sau: Trong không gian W = R² , cho một tập hợp hữu hạn các đa giác lồi O (chướng ngại vật), cần tìm đường đi ngắn nhất từ điểm xuất phát (xs, ys) đến điểm đích (xg, yg) sao cho đường đi chỉ nằm trong không gian khả dụng F = W – O, hay nói cách khác, , tức là không cắt qua các đa giác OO.

Không gian trạng thái là tập hợp các điểm (x, y) ∈ R² nằm trong không gian F. Mặc dù số lượng các điểm trong F là vô hạn, đường đi tối ưu luôn được tạo thành bởi các đỉnh của các đa giác chướng ngại vật và các đoạn thẳng nối điểm đầu và điểm cuối qua không gian F.

1. **Đường đi ngắn nhất Euclid.**

Ta có nhận định sau về đường đi ngắn nhất Euclid đã được chứng minh: Đường đi ngắn nhất từ một đỉnh p nằm trên đa giác đến đỉnh q bất kì nào khác được biểu biễn bởi p = <p, p1, p2, … ,pk ,q>, trong đó mỗi đỉnh p1, p2, …, pk là một đỉnh của một đa giác.

Từ nhận định đó, ta nhận thấy rằng để có thể đi đến điểm cuối bằng cách đi qua đỉnh của các đa giác bằng đường đi Euclid, ta chỉ xem xét các đỉnh của đa giác mà không cần phải xem xét các điểm (x, y) khác nằm trong không gian F. Vậy, không gian trạng thái là S = {(x,y) ∈ R² |(x,y) là một đỉnh của o}. Độ lớn của không gian này là số lượng đỉnh, và không gian hữu hạn này nhỏ hơn rất nhiều so với không gian F.

1. **Đoạn code đã cho đã thiếu hàm**

Xét hình ảnh xuất ra sau khi chạy đoạn code trong file bài tập

A diagram of different shapes and colors

Description automatically generated

Rõ ràng bằng trực quan, ta thấy được rằng đoạn mã đã có vấn đề khi đường đi đã đi xuyên qua các đỉnh của đa giác. Vì vậy, ta cần phải sửa, thay đổi, thêm những hàm cần thiết để giải quyết bài toán.

Ta cần xác định xem, từ một đỉnh A, ta có thể nhìn thấy các đỉnh B nào và có thể đi đến được các đỉnh C nào với B và C là các đỉnh bất kỳ trên đồ thị. Trước hết, ta phải xác định những điểm nhìn thấy được.

A white sheet with black text

Description automatically generated

Xét bài toán tìm những điểm thấy được từ một đỉnh cho trước. Ta thực hiện như sau: Từ đỉnh ta đang xét và các cạnh của đa giác, ta xét lần lượt các cạnh, giả sử cạnh là AB. Những điểm nằm trong cung ASB sẽ là những điểm không nhìn thấy được, và những điểm còn lại sẽ nhìn thấy được. Vì vậy, ta có thể xây dựng thuật toán cho các điểm nhìn thấy được từ một đỉnh cho trước đến các đa giác lân cận.

A diagram of a hexagon with lines and a triangle

Description automatically generated

Tuy vậy, ta lại phát sinh thêm một vấn đề. Thuật toán mà ta xây dựng chỉ cho ta tìm được những điểm nhìn thấy hay có thể đi qua được. Tuy vậy, thực tế rằng không phải tất cả những điểm nhìn thấy được đều có thể đi qua được. Vì vậy, sau khi ta xác định được các điểm nhìn thấy được, ta sẽ loại bỏ đi những đỉnh tạo với đỉnh đang xét một đường chéo của đa giác lân cận.

**Xây dựng thuật toán:**

Sau 5 thuật toán tìm kiếm, ta xây dựng thêm 2 thuật toán nhìn thấy và đi được.

A white paper with black text

Description automatically generated

A white paper with black text

Description automatically generated

**Kết hợp 7 thuật toán, ta sẽ tìm được đường đi đúng cho bài toán với từng thuật toán tìm kiếm.**

**Kết quả sau khi chạy:**

**A\***

**A screenshot of a math graph

Description automatically generated**

**GBFS:**

**A screenshot of a computer

Description automatically generated**

**BFS:**

**A screenshot of a graph

Description automatically generated**

**DFS:**

**A screenshot of a computer

Description automatically generated**

**UCS:**

**A graph with blue lines and a green line

Description automatically generated**

Biểu đồ hiển thị 5 đường đi trong 1 hình:

A diagram of a path

Description automatically generated

**Nhận xét:**

1. Thuật toán A\* cho đường đi tối ưu nhất trong 5 đường đi, vì A\* cân bằng chi phí di chuyển và hàm h(n) để tạo ra đường đi tối ưu nhất có thể.
2. UCS cho đường đi xấu hơn hay không tốt hơn BFS và DFS, vì đường đi của nó dài hơn và ngoằn ngoèo hơn.
3. GBFS cho đường đi tốt hơn UCS và GBFS có đường đi ngắn hơn nhiều so với đường đi của BFS và DFS.

Người báo cáo

Nguyễn Thanh Kiên