

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

ĐỀ TÀI:

ROBOT TÌM ĐƯỜNG

Giảng viên hướng dẫn :

Lê Ngọc Thành

Nhóm sinh viên thực hiện:

Họ và tên	MSSV
Chu Nguyên Đức	1712352
Bùi Chí Dũng	1712364
Nguyễn Công Lý	1712584

TP Hồ Chí Minh, ngày 25 tháng 10 năm 2019

THÔNG TIN NHÓM

MSSV	Họ và tên	Công việc	Đánh giá
1712352	Chu Nguyên Đức	II	100%
1712364	Bùi Chí Dũng	II	100%
1712584	Nguyễn Công Lý	I	100%

Note:

- Thư mục nộp gồm có:

- + 172352_1712364_1712584.pdf: Báo cáo
- + 172352_1712364_1712584.py: File để chạy
- + algorithm.py: File thuật toán
- + Readme.txt: Hướng dẫn chạy

- Đồ án sử dụng thư viện matplotlib để biểu diễn đồ họa. Thư viện này không có sẵn trong Python. Hướng dẫn cài đặt tại [đây](#)

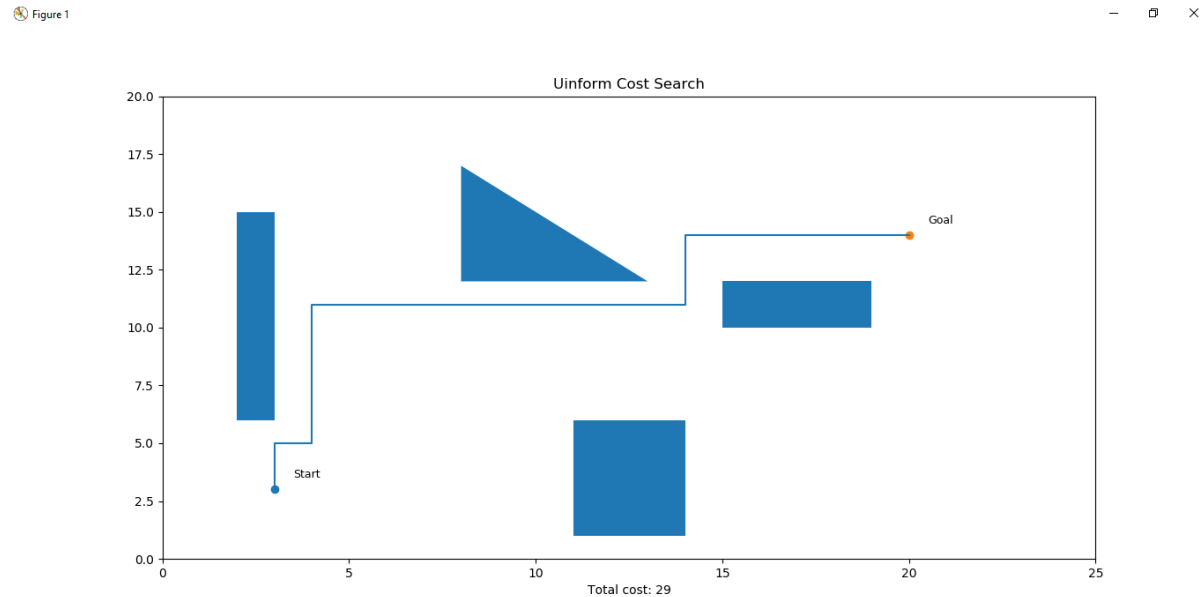
- Không cho phép đi chéo

Mức 1 (40%): Cài đặt thành công 1 thuật toán để tìm đường đi từ S tới G

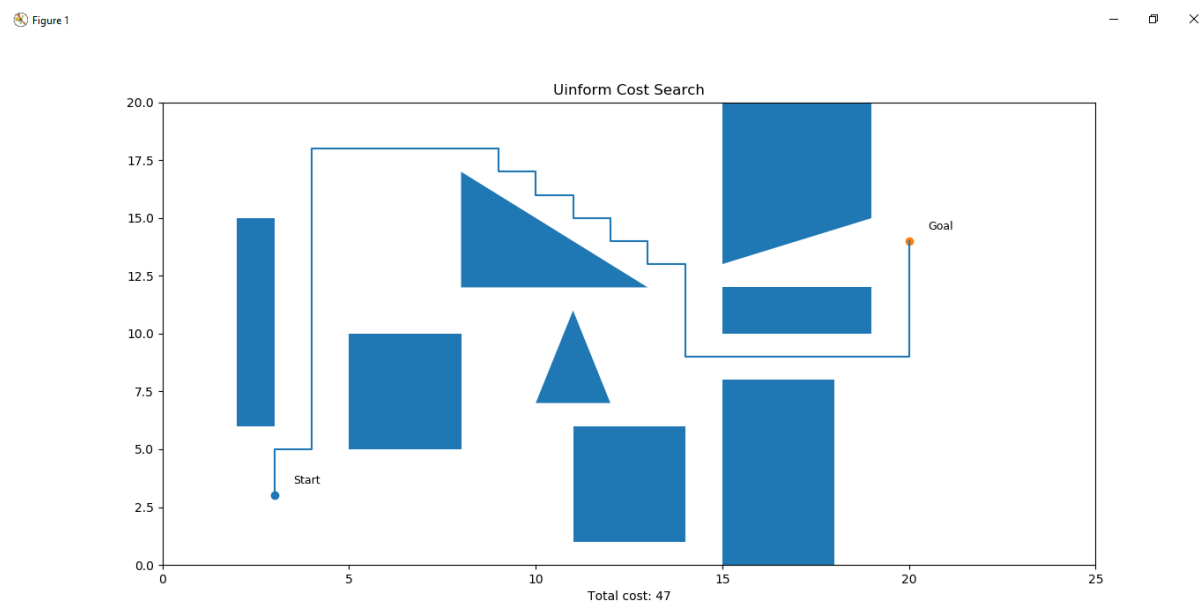
Thuật toán sử dụng: Uniform Cost Search

Chi tiết thuật toán: Việc tìm kiếm bắt đầu tại nút gốc. Việc tìm kiếm tiếp tục bằng cách duyệt các nút tiếp theo với trọng lượng hay chi phí thấp nhất tính từ nút gốc. Các nút được duyệt tiếp tục cho đến khi đến được nút đích cần đến.

Kết quả chạy thử:

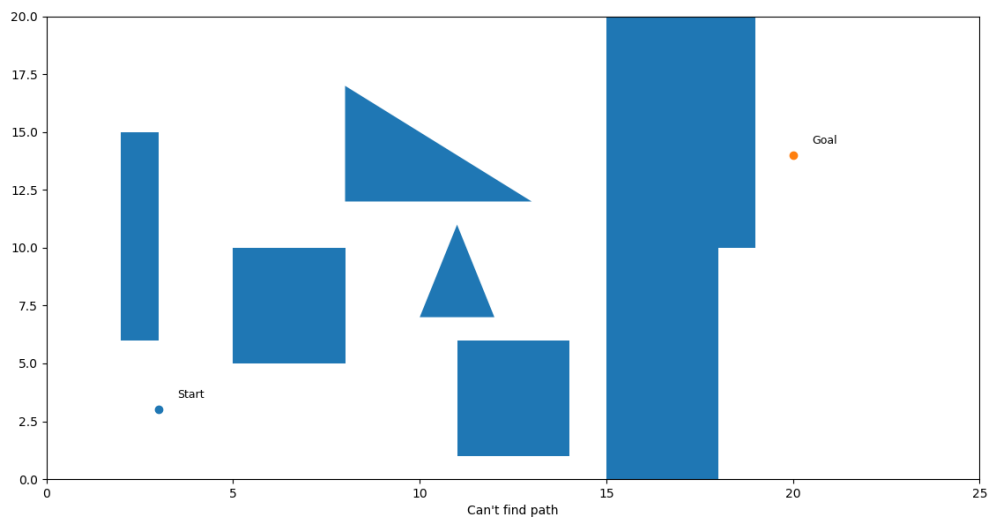


Map 1: Ít vật thể



Map 2: Nhiều vật thể hơn

Figure 1

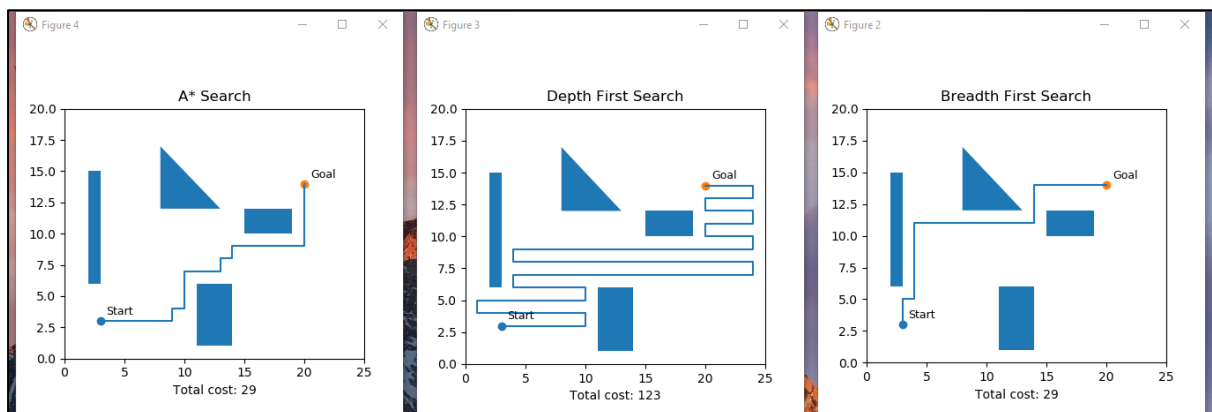


Map 3: Không tìm được đường đi

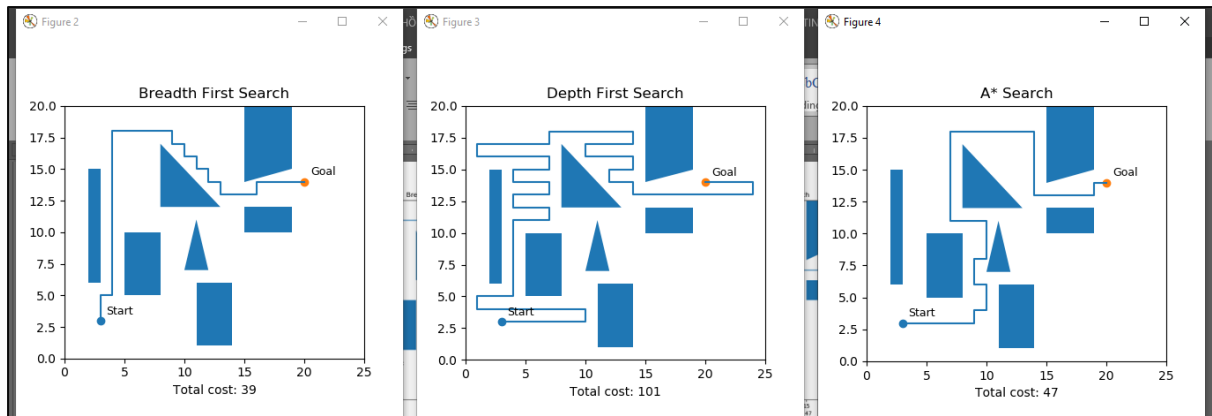
Mức 2 (30%): Cài đặt ít nhất 3 thuật toán khác nhau

Thuật toán sử dụng: Breadth First Search, Depth First Search, A* Search

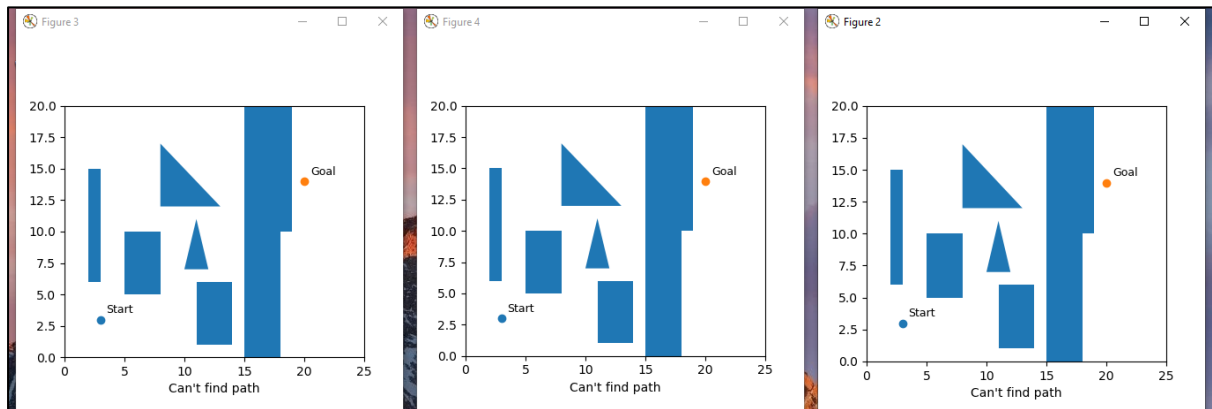
Kết quả chạy thử:



Map 1: Ít vật thể



Map 2: Nhiều vật thể hơn



Map 3: Không tìm được đường đi

So sánh 3 thuật toán

	Breadth First Search	Depth First Search	A* Search
Cách thực hiện	Mở tất cả các điểm con của một điểm	Mở điểm con của một điểm rồi mở điểm con của điểm đó	Mở điểm con có chi phí ít nhất và có ước lượng khoảng cách tới goal ít nhất
Thời gian chạy	Lâu	Lâu	Nhanh
Bộ nhớ	Tốn nhiều bộ nhớ	Tốn nhiều bộ nhớ	Tốn ít bộ nhớ
Tối ưu	Tối ưu để tìm khoảng cách ngắn nhất	Không	Tối ưu
Hoàn thành	Có	Không	Có

Tham khảo:

<https://medium.com/@nicholas.w.swift/easy-a-star-pathfinding-7e6689c7f7b2>

https://en.wikipedia.org/wiki/A*_search_algorithm

<https://www.techiedelight.com/least-cost-path-weighted-digraph-using-bfs/>

<https://www.techiedelight.com/find-path-between-vertices-directed-graph/>