

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN - ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN

LAB 1 – SEARCH

TP. Hồ Chí Minh – 25/10/2020

MỤC LỤC

I. Thông tin cá nhân:	3
II. Nội dung đề án:	4
1. Lý thuyết cơ bản của các thuật toán:	4
a. Tìm kiếm theo chiều rộng - BFS:	4
b. Tìm kiếm theo chiều sâu - DFS:	4
c. Tìm kiếm chi phí đồng nhất - USC:	4
d. Tìm kiếm A^* :	4
2. Sự khác biệt giữa UCS và A^* :	5
3. Chi tiết các thuật toán cài đặt :	6
a. Tìm kiếm theo chiều rộng - BFS:	6
b. Tìm kiếm theo chiều sâu - DFS:	9
c. Tìm kiếm chi phí đồng nhất - UCS:	12
d. Tìm kiếm A^* :	15
III. Mức độ hoàn thành và tự đánh giá:	16
1. Mức độ hoàn thành :	16
2. Tự đánh giá:	16
IV. Tài liệu tham khảo	17

I. Thông tin cá nhân:

- Họ và tên: Lê Hoài Nam
- MSSV: 18120468
- Lớp: 18CTT4

II. Nội dung đề án:

1. Lý thuyết cơ bản của các thuật toán:

a. Tìm kiếm theo chiều rộng - BFS:

- Ý tưởng: Xuất phát từ 1 đỉnh bất kỳ, đi tới tất cả các đỉnh kề của nó, lưu các đỉnh này lại. Tiếp tục đem 1 đỉnh khác (từ tập đỉnh đã được lưu) ra xét và đi cho đến khi không còn đỉnh nào có thể đi. Trong quá trình đi từ đỉnh này sang đỉnh kia, tiến hành lưu lại đỉnh cha của đỉnh kề, để khi đi ngược lại từ đỉnh kết thúc đến đỉnh xuất phát, ta có được đường đi ngắn nhất.
- Độ phức tạp thuật toán: $O(V + E)$ (với V là số đỉnh)

b. Tìm kiếm theo chiều sâu - DFS:

- Ý tưởng: Xuất phát từ 1 đỉnh bất kỳ, đi tới tất cả các đỉnh kề của nó, lưu các đỉnh này lại. Sau đó tiếp tục lấy đỉnh mới nhất vừa được lưu (từ tập đỉnh đã được lưu) ra xét và đi cho đến khi không còn đỉnh nào có thể đi. Trong quá trình đi từ đỉnh này sang đỉnh kia, tiến hành lưu lại đỉnh cha của đỉnh kề, để khi đi ngược lại từ đỉnh Kết Thúc đến đỉnh Xuất Phát, ta có được đường đi cần tìm.
- Độ phức tạp thuật toán: $O(V + E)$ (với V là số đỉnh)

c. Tìm kiếm chi phí đồng nhất - USC:

- Ý tưởng: Việc tìm kiếm bắt đầu tại nút gốc và tiếp tục bằng cách duyệt các nút tiếp theo với trọng số hay chi phí thấp nhất tính từ nút gốc.
- Độ phức tạp thuật toán: $O(b^d)$
Với d , m là độ sâu của cây cần duyệt và mỗi trạng thái khi được phát triển sẽ sinh ra b trạng thái kế (b được gọi là nhân tố nhánh).

d. Tìm kiếm A*:

- Ý tưởng: A* xây dựng tăng dần tất cả các tuyến đường từ điểm xuất phát cho tới khi nó tìm thấy một đường đi chạm tới đích. Tuy nhiên, cũng như tất cả các thuật toán tìm kiếm có thông tin (informed tìm kiếm thuật toán), nó chỉ xây dựng các tuyến đường "có vẻ" dẫn về phía đích. Để biết những tuyến đường nào có khả năng sẽ dẫn tới đích, A* sử dụng một "đánh giá heuristic" về khoảng cách từ điểm bất kỳ cho trước tới đích.
Đánh giá chi phí một nút dựa trên chi phí đi từ nút gốc đến nút đó – $g(n)$, cộng với chi phí đi từ nút đó đến đích – $h(n)$
$$F(n) = g(n) + h(n)$$
- Độ phức tạp thuật toán: tùy theo từng heuristic người dùng cài đặt.

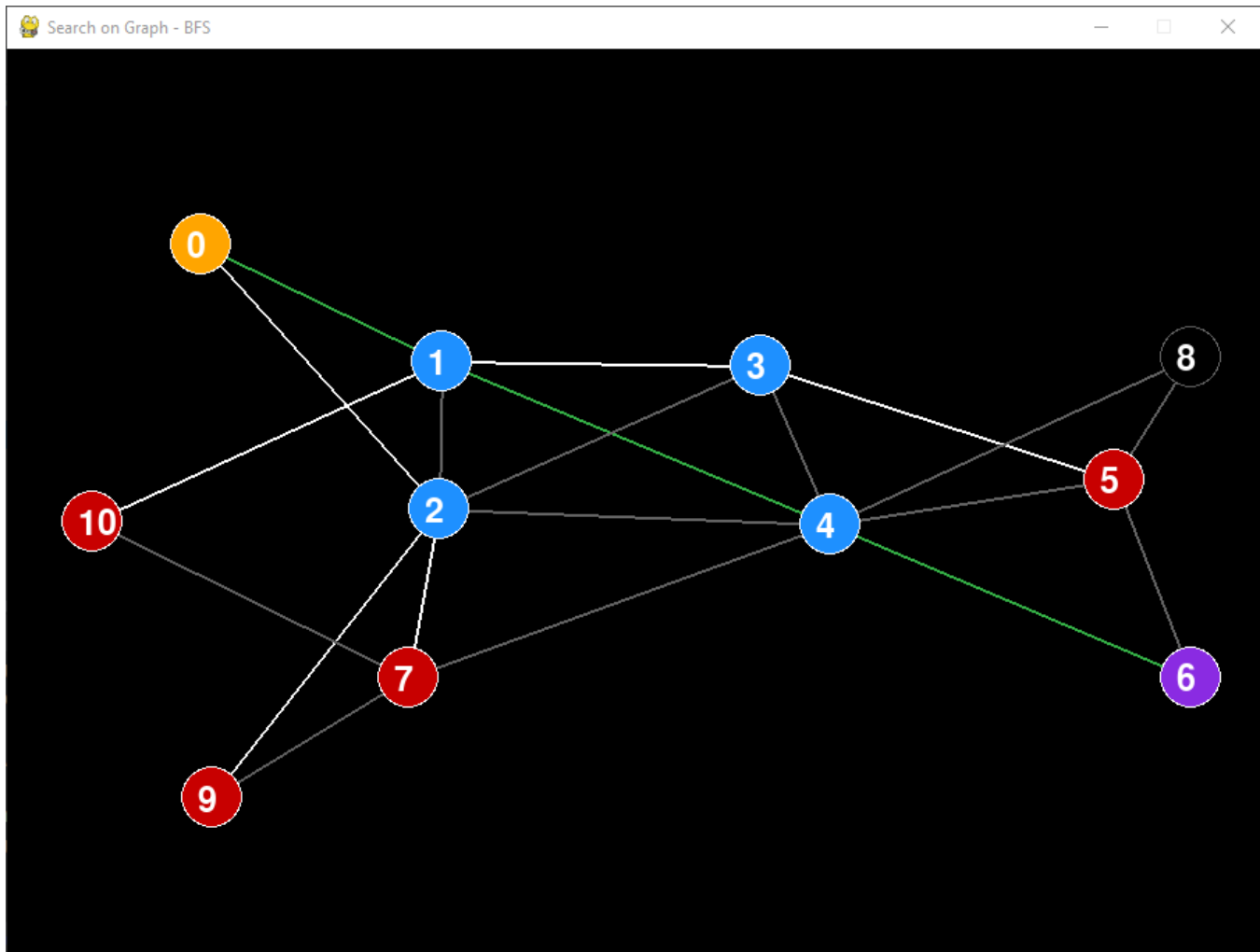
2. Sự khác biệt giữa UCS và A* :

Tiêu chí so sánh	Tìm kiếm chi phí đồng nhất UCS	Tìm kiếm A*
Chiến lược	Tìm kiếm mù, không sử dụng hàm đánh giá	Tìm kiếm kinh nghiệm, sử dụng hàm đánh giá $F(n) = g(n) + h(n)$
Ý tưởng	Tìm đường đi có số bước biến đổi ít nhất	Tìm kiếm đường đi theo chi phí
Chi phí	Tính từ node bắt đầu đến node hiện tại	Tính từ node bắt đầu đến node hiện tại cộng với từ nút đó đến đích
Độ phức tạp theo thời gian	$O(b^d)$	Phụ thuộc hàm heuristic cài đặt
Độ phức tạp theo không gian	$O(b^d)$	Phụ thuộc hàm heuristic cài đặt
Tối ưu	Có nếu $cost > 0$	Để tối ưu thì hàm $h(n)$ phải là hàm heuristic chấp nhận được. Hàm heuristic chấp nhận được khi nó không bao giờ ước lượng quá chi phí để đến đích thực sự
Ghi chú	Khi đồ thị có chi phí ở mỗi bước là như nhau thì thuật toán trở thành phương pháp tìm kiếm theo chiều rộng.	

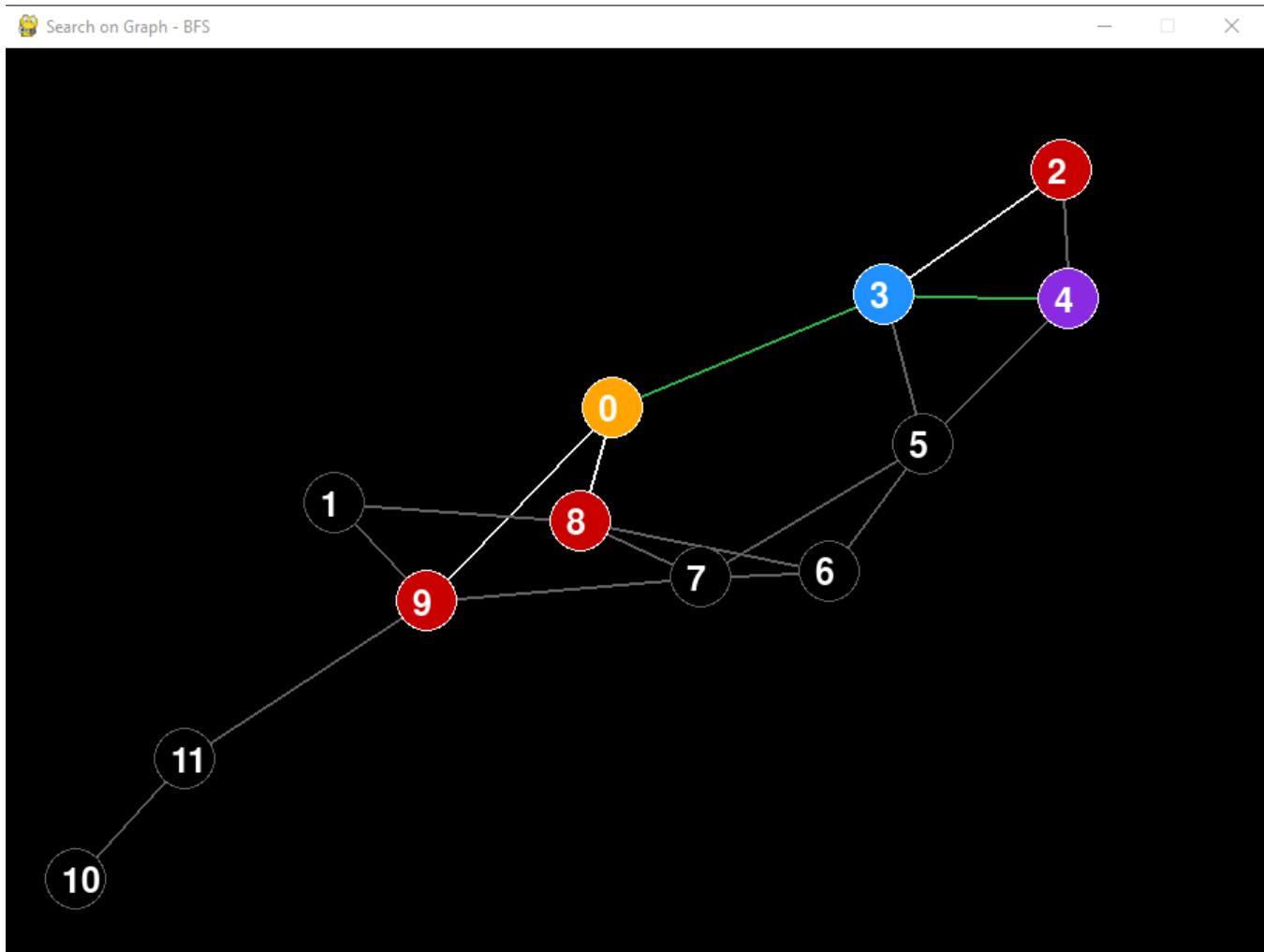
3. Chi tiết các thuật toán cài đặt :

a. *Tìm kiếm theo chiều rộng - BFS:*

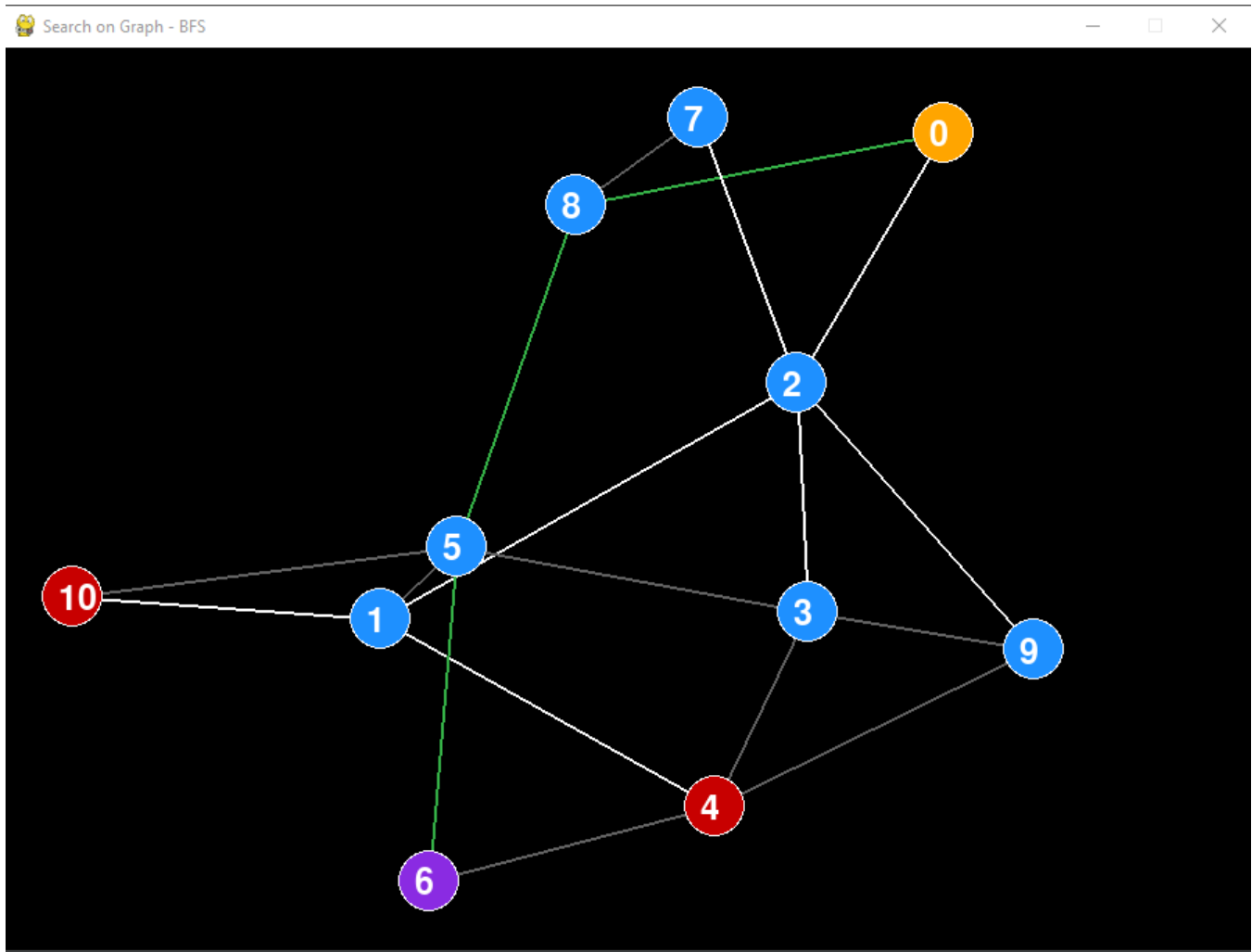
- Test case 1:



- Test case 2:



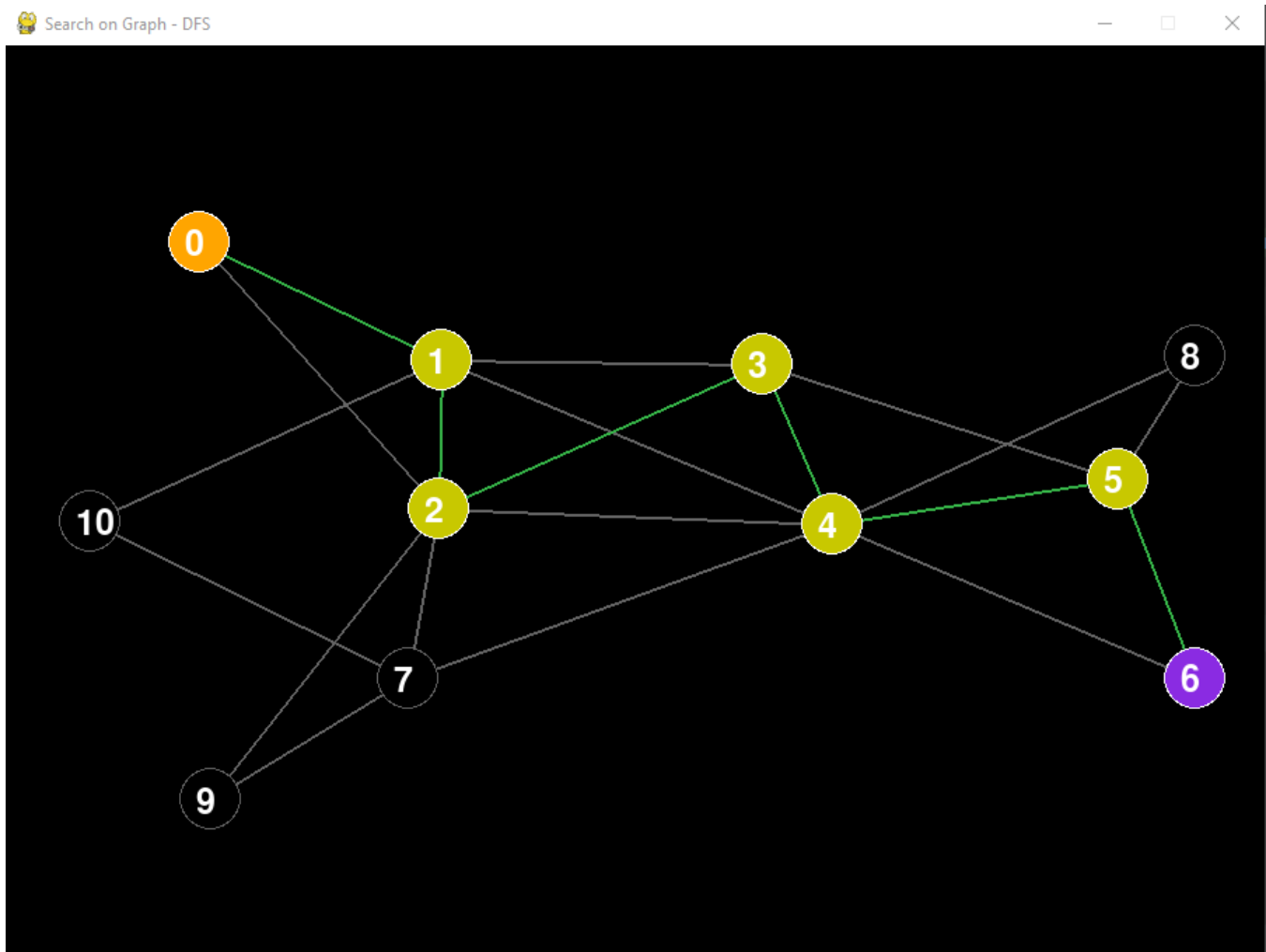
- Test case 3:



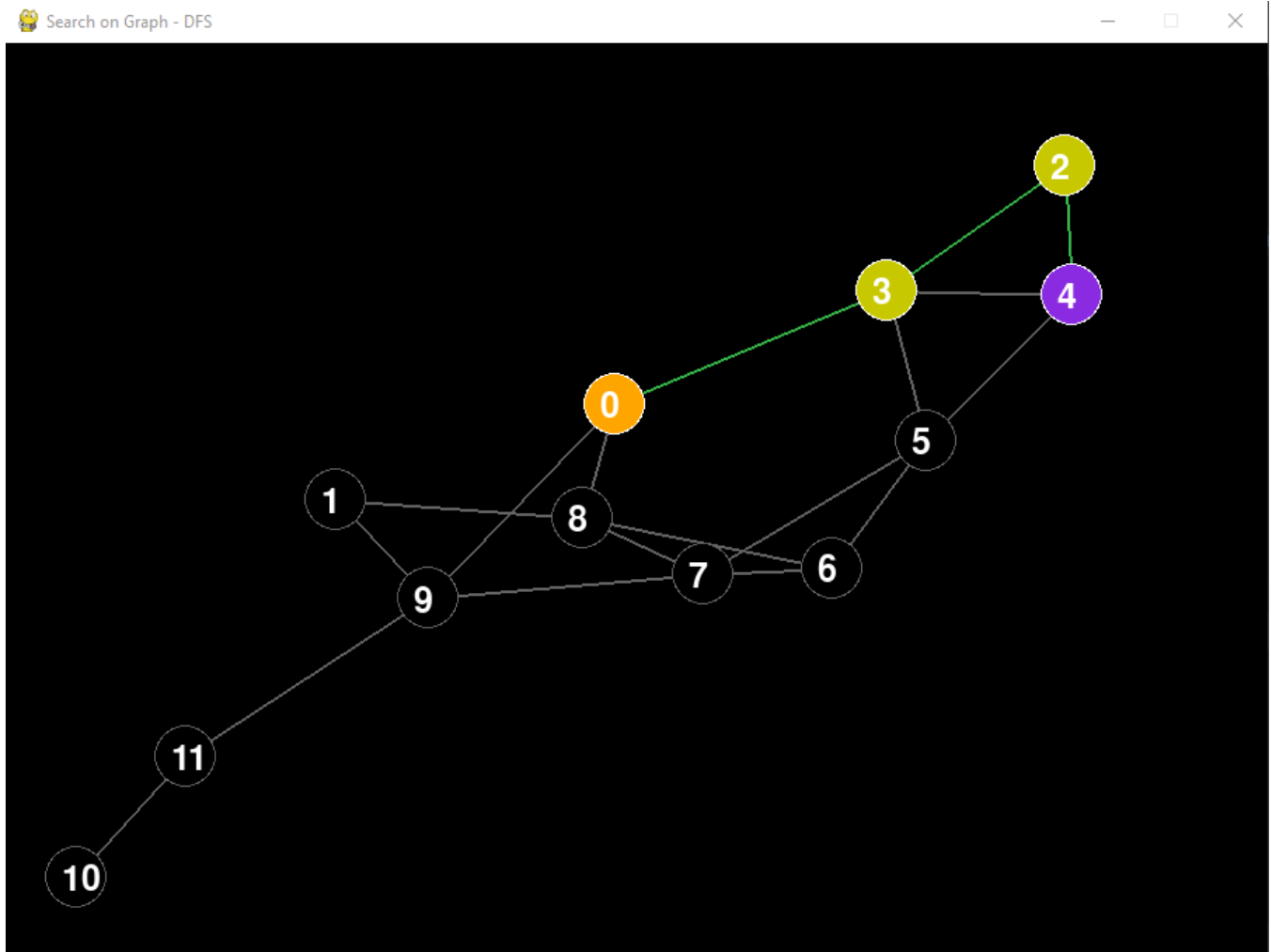
- Nhận xét: Kỹ thuật tìm kiếm rộng là kỹ thuật vét cạn không gian trạng thái bài toán vì vậy sẽ tìm được lời giải nếu có và đường đi tìm được đi qua ít đỉnh nhất.

b. Tìm kiếm theo chiều sâu - DFS:

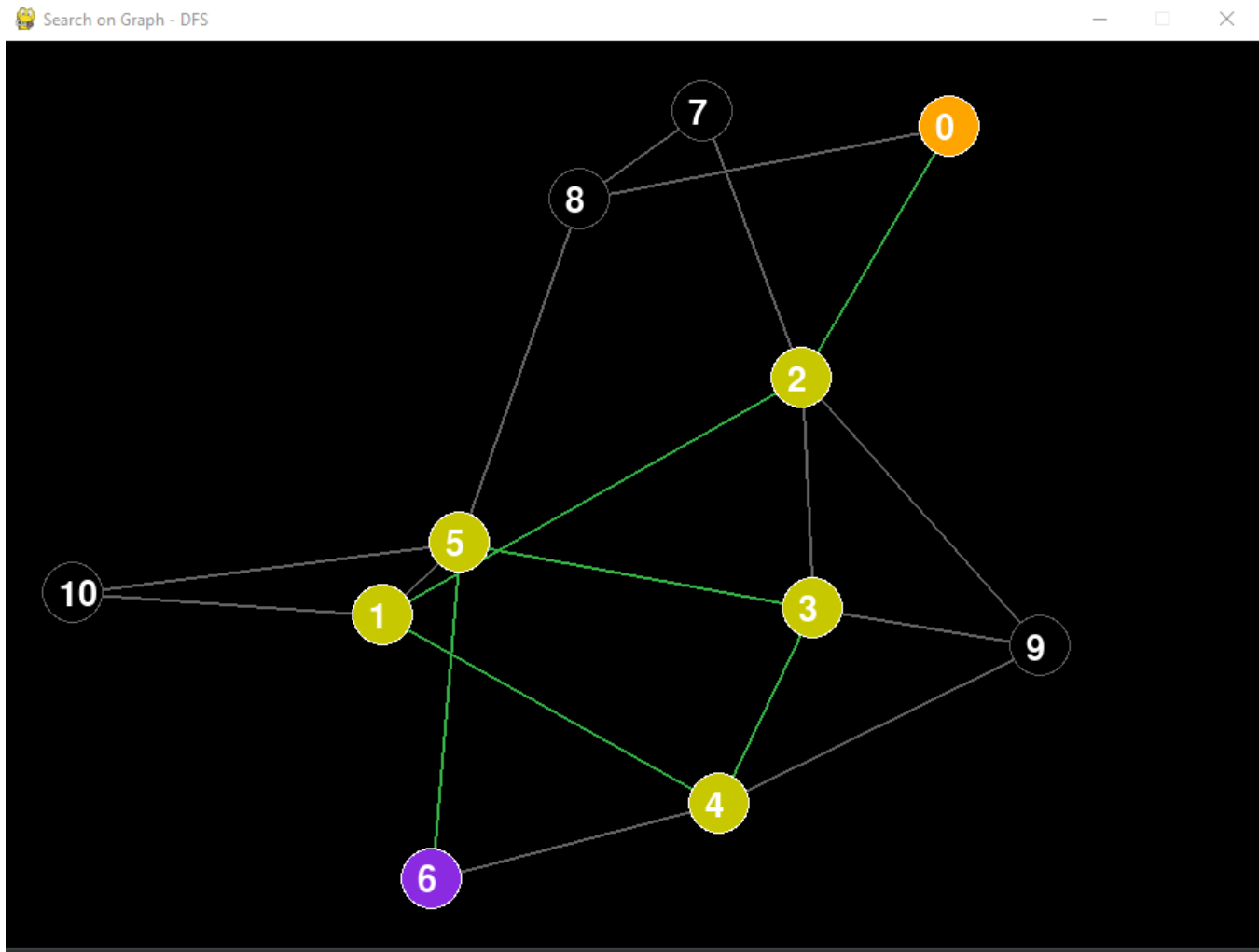
- Testcase 1:



- Testcase 2:



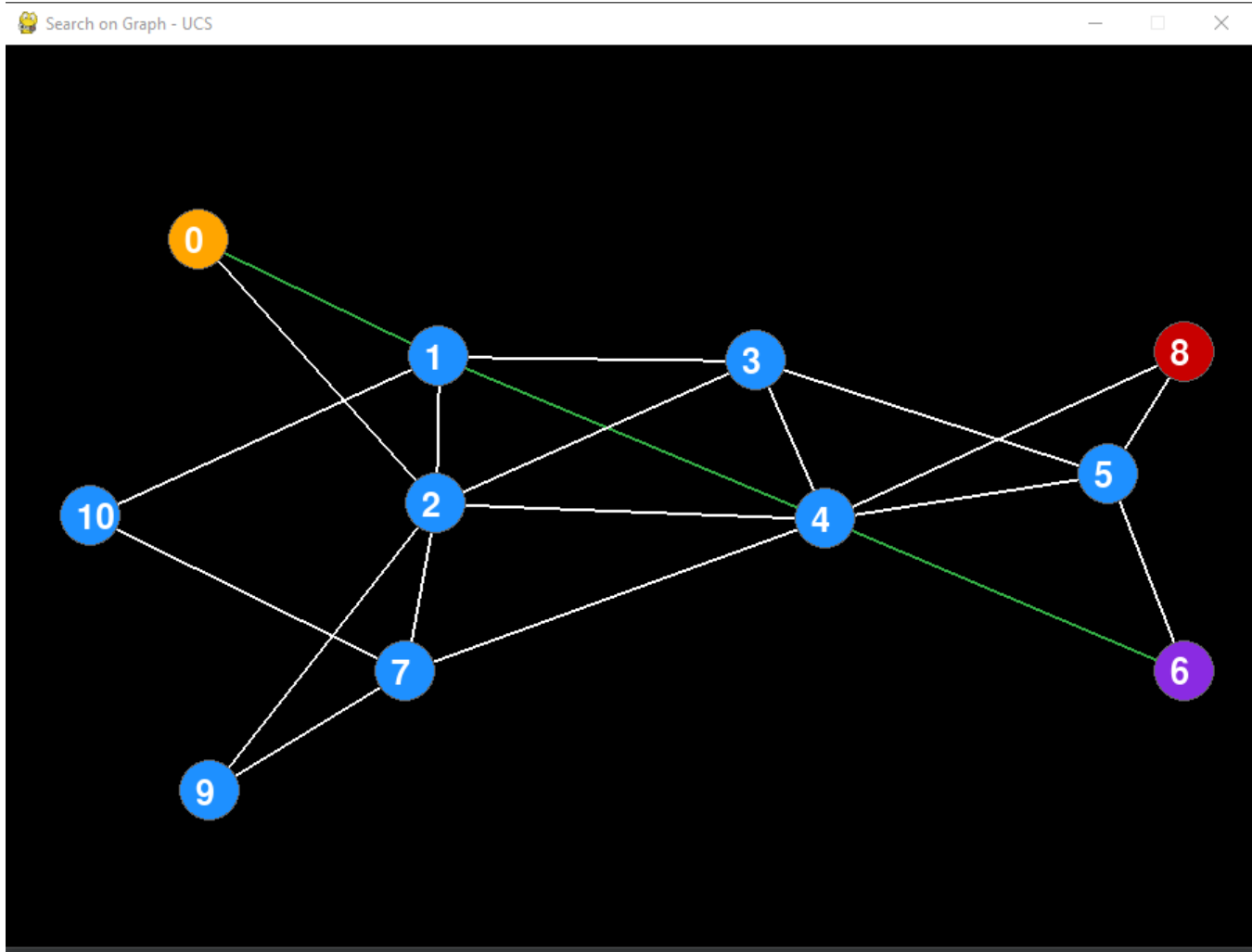
- Testcase 3:



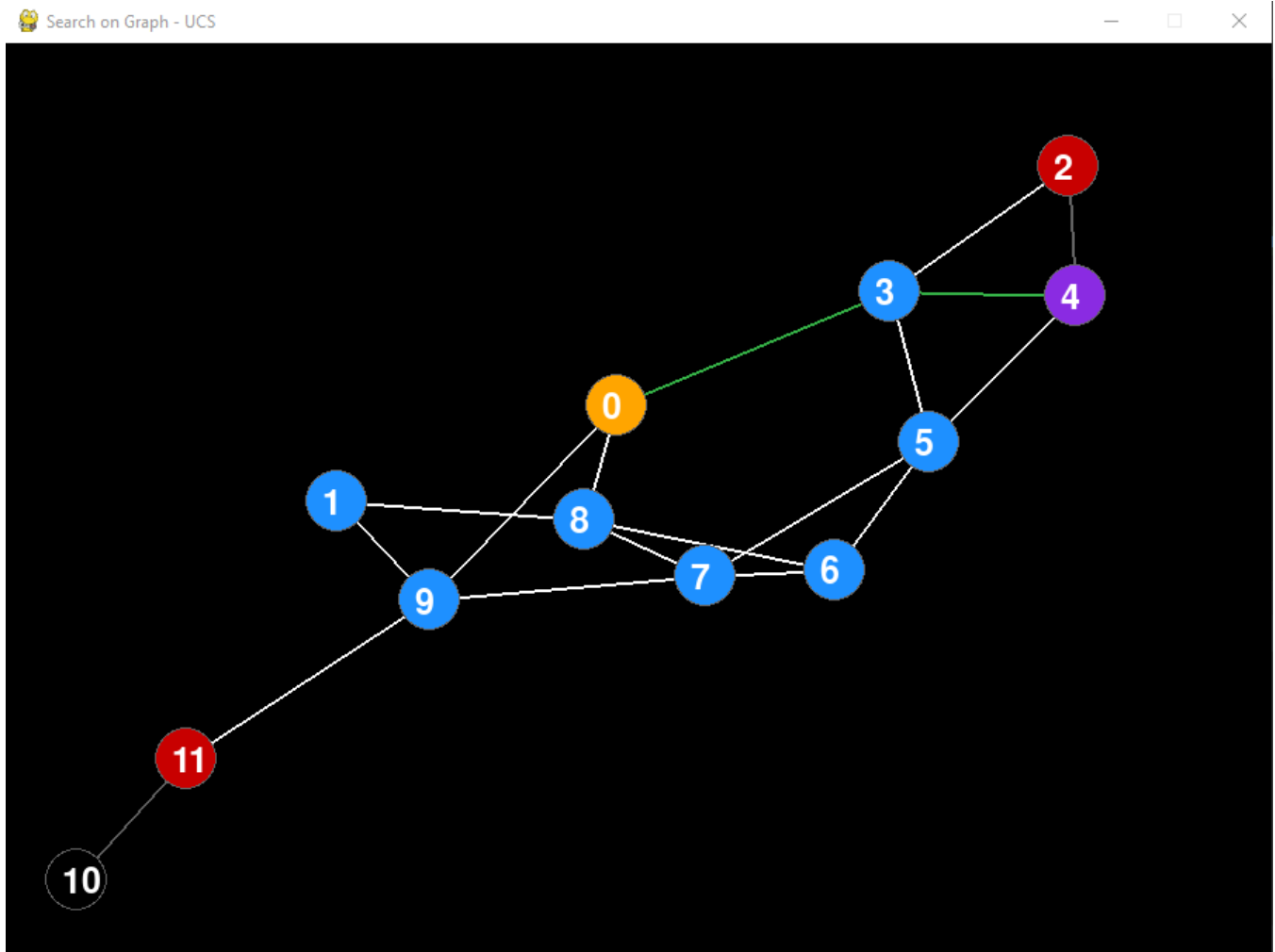
- Nhận xét: Xét duyệt tất cả các đỉnh để trả về kết quả, nếu số đỉnh là hữu hạn thuật toán chắc chắn tìm ra kết quả.

c. Tìm kiếm chi phí đồng nhất - UCS:

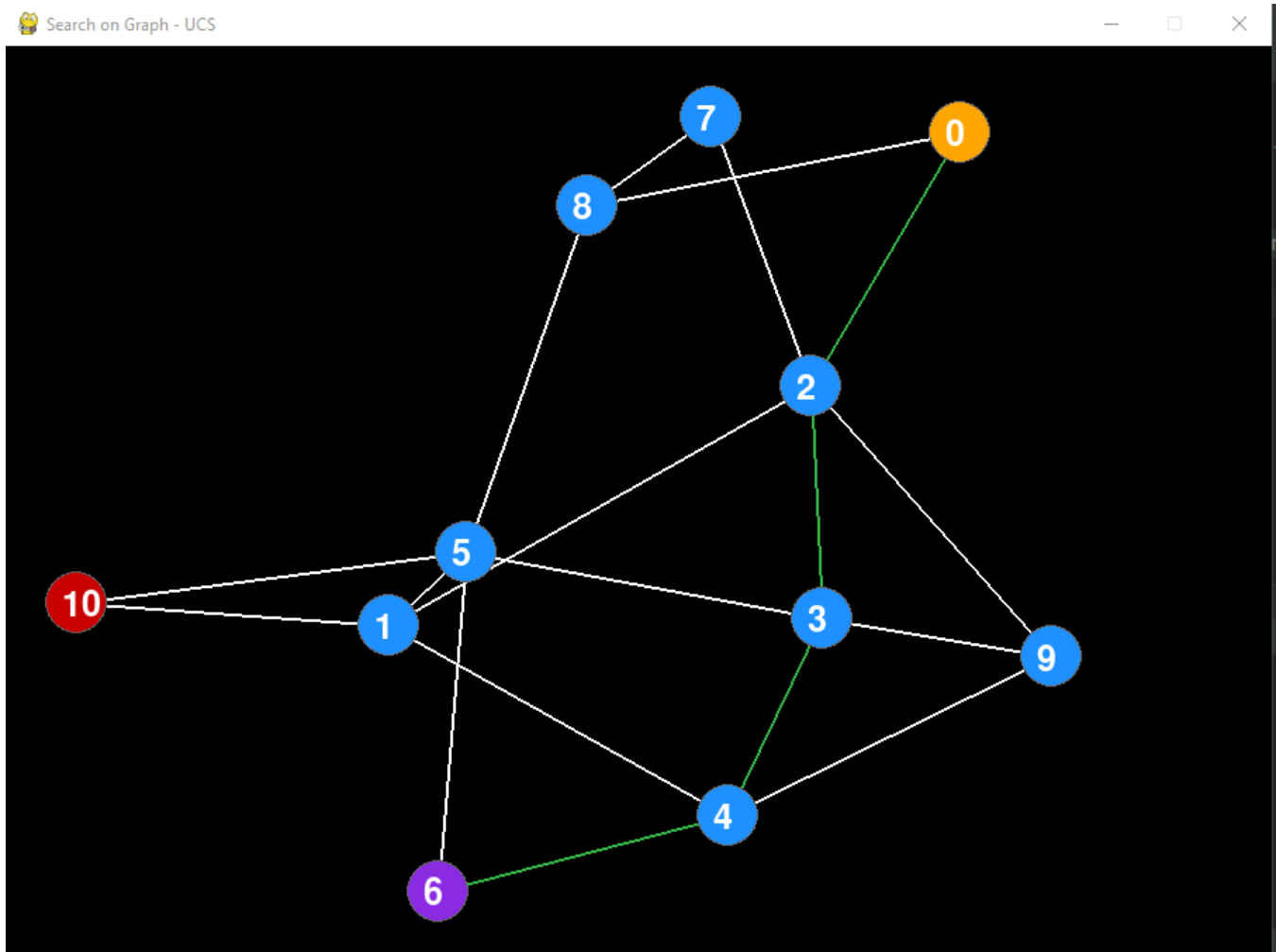
- Testcase 1:



- Testcase 2:



- Testcase 3:



- Nhận xét: Khi đồ thị có chi phí ở mỗi bước là như nhau thì thuật toán trở thành phương pháp tìm kiếm theo chiều rộng.

d. Tìm kiếm A:*

- Testcase 1:
 - Testcase 2:
 - Testcase 3:
-
- Nhận xét: Một thuật giải linh động, tổng quát, trong đó hàm chứa cả tìm kiếm chiều sâu, tìm kiếm chiều rộng và những nguyên lý Heuristic khác. Nhanh chóng tìm đến lời giải với sự định hướng của hàm Heuristic. A* rất linh động nhưng vẫn gặp một khuyết điểm cơ bản đó là tốn khá nhiều bộ nhớ để lưu lại những trạng thái đã đi qua.

III. Mức độ hoàn thành và tự đánh giá:

1. Mức độ hoàn thành :

STT	Thuật toán	Mức độ hoàn thành
1	Tìm kiếm theo chiều rộng - BFS	100%
2	Tìm kiếm theo chiều sâu - DFS	100%
3	Tìm kiếm chi phí đồng nhất - USC	100%
4	Tìm kiếm A*	40%

2. Tự đánh giá:

- Điểm tự đánh giá : 7.5đ

IV. Tài liệu tham khảo

- Sách Cơ sở trí tuệ nhân tạo – Lê Hoài Bắc – Tô Hoài Việt -Trường Đại học Khoa học Tự Nhiên
- Nhập môn Cấu Trúc Dữ Liệu và Giải Thuật, Dương Anh Đức, Đại Học Khoa Học Tự Nhiên TP.HCM
- Wikipedia

https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_theo_chi%E1%BB%81u_s%C3%A2u

https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_theo_chi%E1%BB%81u_r%E1%BB%99ng

https://vi.wikipedia.org/wiki/Gi%E1%BA%A3i_thu%E1%BA%ADt_t%C3%ACm_ki%E1%BA%BFm_A*

- Medium

https://medium.com/@manhtandthu/thu%E1%BA%ADt-to%C3%A1n-t%C3%ACm_ki%E1%BA%BFm-theo-chi%E1%BB%81u-r%E1%BB%99ng-73f6f17c06eb

<https://medium.com/@manhtandthu/thu%E1%BA%ADt-to%C3%A1n-DFS-c-python-df109b51e129>

https://medium.com/@manhtandthu/thu%E1%BA%ADt-to%C3%A1n-t%C3%ACm_ki%E1%BA%BFm-theo-chi%E1%BB%81u-r%E1%BB%99ng-73f6f17c06eb

- Viblo.asia

https://viblo.asia/p/cac-thuat-toan-co-ban-trong-ai-phan-biet-best-first-search-va-uniform-cost-search-ucs-Eb85omLWZ2G#_thuath-toan-ucs-2

- Stackoverflow

<https://stackoverflow.com/questions/44849517/what-is-the-time-complexity-of-a-search>

- Github

- Geeksforgeeks