

ĐẠI HỌC QUỐC GIA HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Báo cáo
Cài đặt thuật toán DFS, BFS và UCS
để chơi game Sokoban

GVHD:	Lương Ngọc Hoàng
Họ tên SV:	Đặng Thanh Ngân
MSSV:	22520929
Lớp:	CS106.O22

Mục lục

I. Mô tả Sokoban	3
II. Áp dụng giải thuật DFS, BFS, UCS trong Sokoban	3
1. Bảng thống kê	3
2. Nhận xét.....	4
2.1. Nhận xét kết quả	4
2.2. Thuật toán tốt nhất trong bài toán	5
2.3. Bản đồ khó giải nhất	6

I. Mô tả Sokoban

Không gian trạng thái: Vị trí người chơi và các hộp.

Trạng thái khởi đầu: Vị trí của ban đầu của người chơi, của các hộp.

Trạng thái kết thúc: Các hộp được đặt đúng vị trí đích.

Hàm tiền triển (successor function): Cập nhật vị trí của người chơi, các hộp chưa được đặt vào đúng vị trí.

Các hành động hợp lệ:

1. Di chuyển lên: Người chơi có thể di chuyển lên trên nếu như ô đó trống (không phải tường hay hộp).
2. Di chuyển xuống: Người chơi có thể di chuyển xuống nếu như ô đó trống.
3. Di chuyển sang phải: Người chơi có thể di chuyển sang phải nếu như ô đó trống.
4. Di chuyển sang trái: Người chơi có thể di chuyển sang trái nếu như ô đó trống.
5. Đẩy lên trên: Nếu có một ô cạnh người chơi và có một ô trống ở hướng phía trên ô thì người chơi sẽ đẩy ô đó lên một ô.
6. Đẩy xuống: Nếu có một ô cạnh người chơi và có một ô trống ở hướng phía dưới ô thì người chơi sẽ đẩy ô đó lên một ô.
7. Đẩy sang phải: Nếu có một ô cạnh người chơi và có một ô trống ở hướng phía bên phải ô thì người chơi sẽ đẩy ô đó lên một ô.
8. Đẩy sang trái: Nếu có một ô cạnh người chơi và có một ô trống ở hướng phía bên phải ô thì người chơi sẽ đẩy ô đó lên một ô.

II. Áp dụng giải thuật DFS, BFS, UCS trong Sokoban

1. Bảng thống kê

Số bước đi và thời gian (s) của từng thuật toán

Thuật toán Level	DFS	BFS	UCS
1	79	12	12
	0.06	0.08	0.07
2	24	9	9
	0.00	0.01	0.01
3	403	15	15
	0.26	0.16	0.10

4	27	7	7
	0.00	0.01	0.01
5	Chương trình bị đứng	20	20
		252.74	187.92
6	55	19	19
	0.01	0.02	0.01
7	707	21	21
	0.61	0.92	0.68
8	323	97	97
	0.09	0.18	0.19
9	74	8	8
	0.30	0.1	0.01
10	37	34	33
	0.01	0.02	0.02
11	36	34	34
	0.02	0.02	0.02
12	109	23	23
	0.16	0.08	0.09
13	185	31	31
	0.2	0.13	0.17
14	865	23	23
	4.36	2.87	3.32
15	291	105	105
	0.15	0.25	0.32
16	Chương trình bị đứng	34	34
		27.47	25.12
17	Chương trình bị đứng	Không có lời giải	Phần mềm tự thoát vì lỗi Index
		34.06	
18	Chương trình bị đứng	Chương trình bị đứng	Chương trình bị đứng

2. Nhận xét

2.1. Nhận xét kết quả

-DFS có đường đi dài hơn BFS và UCS. Với những bản đồ ít phức tạp thời gian tìm lời giải của DFS nhanh hơn hai thuật toán còn lại, nhưng nó sẽ lâu hơn khi độ phức tạp tăng nhiều lần như level 9, 10, 12, 13, 14.

-BFS và UCS có số bước để đẩy các thùng tới đích bằng nhau vì chi phí khi di chuyển các bước đều giống nhau. Nhưng thời gian thực hiện giải thuật BFS lâu hơn UCS vì nó tìm tất cả những đường có thể có theo từng cấp độ, điều này yêu cầu nhiều bộ nhớ và thời gian để xử lý.

-Có những bản đồ BFS và UCS có thể giải được nhưng DFS không giải được như bản đồ ở level 5 và 16. DFS tìm lời giải theo chiều sâu, có thể không thể tìm được lời giải cho bài toán. BFS và UCS chắc chắn sẽ tìm được lời giải và tối ưu nhưng đòi hỏi bộ nhớ và thời gian.

-Với những bản đồ không có lời giải level 17 thì DFS sẽ đào sâu nhất có thể để tìm đáp án nên chương trình sẽ không phát hiện là bài toán không có lời giải. Còn BFS kiểm tra theo từng cấp độ và tới cấp cuối cùng thuật toán có thể thông báo chương trình không có lời giải. UCS sẽ tìm kiếm theo chi phí cho tới khi tràn bộ nhớ.

-Với những màn phức tạp, nhiều thùng như level 18, thì cả 3 thuật toán đều không tìm ra được lời giải.

2.2. Thuật toán tốt nhất trong bài toán này

Việc chọn thuật toán tốt nhất để giải Sokoban phụ thuộc vào một số yếu tố, bao gồm các đặc điểm cụ thể của các trường hợp, tài nguyên tính toán sẵn có và các yêu cầu của ứng dụng. Mỗi DFS, BFS và UCS đều có điểm mạnh và điểm yếu, vì vậy lựa chọn tốt nhất có thể khác nhau tùy theo ngữ cảnh.

Đây là điểm mạnh và điểm yếu của từng thuật toán:

	Điểm mạnh	Điểm yếu
DFS	DFS tiết kiệm bộ nhớ và thực hiện tương đối đơn giản. Nó phù hợp với các câu đố có hệ số phân nhánh và tài nguyên bộ nhớ hạn chế.	DFS không phải lúc nào cũng có thể tìm ra giải pháp tối ưu và có thể gặp khó khăn khi khám phá các nhánh sâu, dẫn đến kém hiệu quả
BFS	BFS hoàn chỉnh và đảm bảo tìm ra con đường ngắn nhất dẫn đến giải pháp. Nó phù hợp cho các câu đố trong đó tính tối ưu là rất quan trọng.	BFS có thể tốn nhiều bộ nhớ do cần lưu trữ tất cả các nút ở mỗi cấp của cây tìm kiếm. Nó cũng có thể chậm đối với các câu đố có hệ số phân nhánh cao.
UCS	UCS tối ưu và đảm bảo tìm ra giải pháp với chi phí tối thiểu.	UCS có thể tốn nhiều bộ nhớ và có thể có độ phức tạp về thời gian cao hơn so

	Nó phù hợp cho các câu đố trong đó việc giảm thiểu chi phí là quan trọng.	với BFS và DFS, đặc biệt nếu hàm chi phí không được triển khai hiệu quả. Xét về thời gian và độ tiết kiệm bước đi thì UCS chiếm ưu thế, cũng như tiết kiệm không gian bộ nhớ so với BFS
--	---	---

Đối với các câu đố Sokoban có hệ số phân nhánh thấp và hạn chế về bộ nhớ, DFS có thể mang lại sự cân bằng tốt giữa tính hiệu quả và tính đơn giản.

Nếu tính tối ưu là rất quan trọng và tài nguyên bộ nhớ không phải là vấn đề đáng lo ngại thì BFS sẽ là một lựa chọn phù hợp.

Đối với các câu đố Sokoban trong đó việc giảm thiểu số lần di chuyển hoặc hành động đẩy là quan trọng, UCS có thể là lựa chọn tốt nhất do tính tối ưu về mặt chi phí.

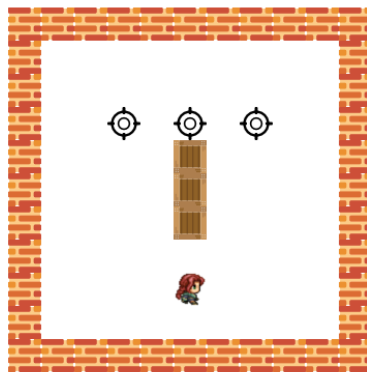
Từ bảng thống kê trên, ta thấy thuật toán UCS có lời giải cho hầu hết cho các bản đồ với số bước đi ít nhất và runtime nhỏ hơn BFS, và ít chiếm bộ nhớ hơn BFS. Khi bài toán không có lời giải UCS thoát chương trình nhanh hơn BFS. Có thể thấy thuật toán UCS là thuật toán tốt ưu nhất cho bài toán Sokoban này.

2.3. Bản đồ khó giải nhất

Level 5

Restart level

Undo the last move
Auto



Trong các bản đồ, bản đồ level 5 yêu cầu nhiều thời gian để các thuật toán để xử lý.

Một số lý do chủ yếu là:

- Bản đồ có nhiều không gian, ít chướng ngại vật nên các thuật toán như BFS, UCS phải xử lý nhiều node hơn. Ở một số cấp độ Sokoban, có thể có một số lượng lớn các bước di chuyển có thể có ở mỗi trạng thái, dẫn đến hệ số phân nhánh cao trong cây tìm

kiếm. BFS khám phá tất cả các bước di chuyển có thể có theo cấp độ và nếu hệ số phân nhánh cao, số lượng nút cần khám phá có thể nhanh chóng trở nên không thể quản lý được.

-Nhiều hộp cũng là lý do khiến các thuật toán chạy chậm vì phải xử lý từng hộp.

-Bản đồ này DFS không giải được vì nhiều đường làm thuật toán DFS dẫn các hộp đến vị trí chết không thể di chuyển đi được nữa. Các câu đố về Sokoban thường chứa đựng những ngõ cụt hoặc những con đường không dẫn đến lời giải. DFS có thể gặp khó khăn khi khám phá những ngõ cụt này, lãng phí thời gian và tài nguyên tính toán.

Ngoài ra, nếu câu đố Sokoban chứa các đường dẫn vô hạn, DFS có thể bị mắc kẹt trong một vòng lặp vô hạn và không thể kết thúc.

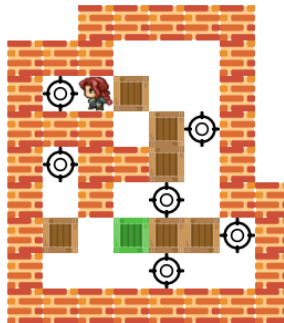
-Các cấp độ Sokoban thường chứa đựng những ngõ cụt hoặc những con đường không dẫn đến giải pháp. BFS khám phá tất cả các đường dẫn có thể như nhau cho đến khi tìm ra giải pháp, điều đó có nghĩa là nó có thể lãng phí thời gian để khám phá những ngõ cụt hoặc các đường dẫn dư thừa này trước khi tìm ra giải pháp chính xác.

Theo sau đó là bản đồ level 16:

Level 16

Restart level

Undo the last move
Auto



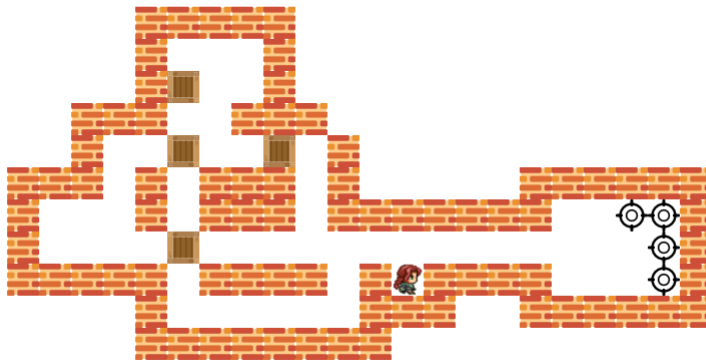
-Bản đồ này được thiết kế một các phức tạp và gồm nhiều hộp (7 hộp) nên tốn thời gian và số bước đi nhiều hơn. Một số cấp độ Sokoban được thiết kế đặc biệt phức tạp, đòi hỏi các chuỗi di chuyển phức tạp và định vị chính xác các hộp để giải. BFS khám phá toàn bộ không gian tìm kiếm một cách thấu đáo, do đó, ngay cả trong các tình huống tồn tại giải pháp tối ưu, độ phức tạp tuyệt đối của câu đố có thể khiến việc tìm kiếm nó tốn thời gian.

-Level 17 không có lời giải làm cho các thuật toán phải chạy đến khi hết node trong hàng đợi của chúng. Đến khi hết bộ nhớ thì mới thoát.

Level 17

Restart level

Undo the last move
Auto



Trong các bản đồ, Level 18 là bản đồ khó nhất, vì có nhiều thùng cần được đặt vào đúng vị trí khiến tốn nhiều thời gian để xử lý. Thêm vào đó có nhiều tầng khiến cho việc tìm đường đi dẫn tới các điểm chết. Cả 3 thuật toán đều không có lời giải một cách nhanh chóng, cho thấy được độ phức tạp của bài toán.

Level 18

Restart level

Undo the last move
Auto

