

ĐỒ ÁN CÁ NHÂN MÔN NHẬP MÔN TRÍ TUỆ NHÂN TẠO

Giảng viên hướng dẫn: TS.Phan Thị Huyền Trang

Sinh viên thực hiện: Võ Lê Khánh Duy

Mã số sinh viên: 2311096

Mã lớp học: ARIN330585_04

Học kỳ II năm học 2024-2025

MỤC LỤC

- 1. Giới thiệu bài toán
- 2. Mục tiêu
- 3.1. Nội dung
 - 3. Một số thuật toán sử dụng
 - 3.1. Tìm kiếm không có thông tin
 - Các thành phần chính của bài toán tìm kiếm
 - Thuật toán BFS
 - Thuật toán DFS
 - Thuật toán IDS
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
 - 3.2. Tìm kiếm có thông tin
 - Các thành phần chính của bài toán tìm kiếm
 - Thuật toán USC - Uniform Cost Search
 - Thuật toán Greedy
 - Thuật toán A
 - Thuật toán Iterative deepening a
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
 - 3.3. Thuật toán tìm kiếm cục bộ
 - Các thành phần chính của bài toán tìm kiếm
 - Thuật toán Simple hill climbing
 - Thuật toán Steepest ascent hill climbing
 - Thuật toán Stochastic hill climbing
 - Thuật toán Beam Search
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
 - 3.4. Thuật toán tìm kiếm trong môi trường phức tạp Các thành phần chính của bài toán tìm kiếm
 - Thuật toán tìm kiếm AND - OR graph search
 - Thuật toán tìm kiếm không có sự quan sát

- Thuật toán tìm kiếm có sự quan sát một phần
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
 - 3.5. Thuật toán tìm kiếm có ràng buộc Các thành phần chính của bài toán tìm kiếm
 - Thuật toán kiểm thử
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
 - 3.6. Thuật toán tìm kiếm học tăng cường Các thành phần chính của bài toán tìm kiếm
 - Thuật toán tìm kiếm Q Learning
 - Hình ảnh so sánh hiệu suất của các thuật toán
 - Nhận xét về hiệu suất của các thuật toán trong nhóm
- 4. Kết luận
 - 5. Tài liệu tham khảo

NỘI DUNG CHÍNH

1. Giới thiệu bài toán

Bài toán 8 puzzle có trạng thái ban đầu là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống).

Bài toán 8 puzzle có trạng thái mục tiêu cũng là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống).

Yêu cầu đặt ra của bài toán là di chuyển số 0 (ô trống) theo 1 trong các hướng (hành động) lên, xuống, trái, phải sao cho từ trạng thái ban đầu đạt được trạng thái mục tiêu qua 1 số bước chuyển đổi ô trống.

Tùy thuộc vào thuật toán sử dụng, ta có thể tìm ra lời giải là đường đi từ trạng thái ban đầu đến trạng thái mục tiêu trải qua số bước, thời gian giải, không gian trạng thái mở rộng có thể khác nhau, thậm chí có một số trường hợp một toán có thể không tìm ra được lời giải.

2. Mục tiêu

Áp dụng các thuật toán tìm kiếm trí tuệ nhân tạo đã học ở lớp vào giải quyết một vấn đề cụ thể. Từ đó đưa ra những đánh giá về hiệu suất, hiệu quả của từng loại thuật toán về mặt thời gian, không gian trong từng trường hợp cụ thể. Từ đây, hoàn toàn có thể áp dụng các thuật toán này vào các bài toán thực tiễn như tìm đường đi giữa các thành phố, tìm kiếm thông tin nhanh chóng hơn trong kho dữ liệu lớn, lập lịch, học máy...

[1]

3. Một số thuật toán sử dụng

Xét trạng thái đầu vào và trạng thái mục tiêu như hình sau:

Start state matrix:			End state matrix:		
2	6	5	1	2	3
0	8	7	4	5	6
4	3	1	7	8	0

3.1. Tìm kiếm không có thông tin

Các thành phần chính của bài toán tìm kiếm

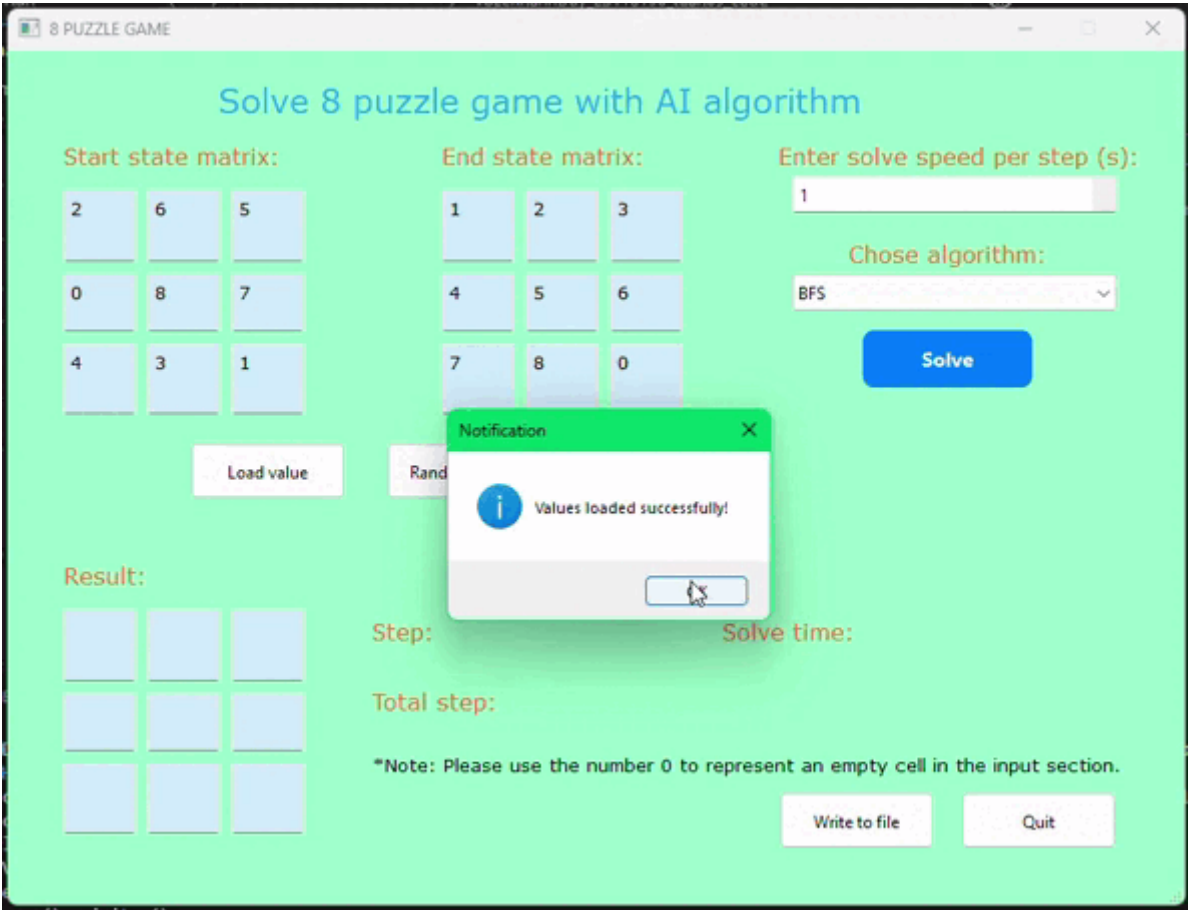
Bài toán 8 puzzle có trạng thái ban đầu là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu vào bài toán cần giải quyết.

Bài toán 8 puzzle có trạng thái mục tiêu cũng là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu ra của bài toán (là trạng thái muốn có sau khi thực hiện các hành động từ trạng thái ban đầu).

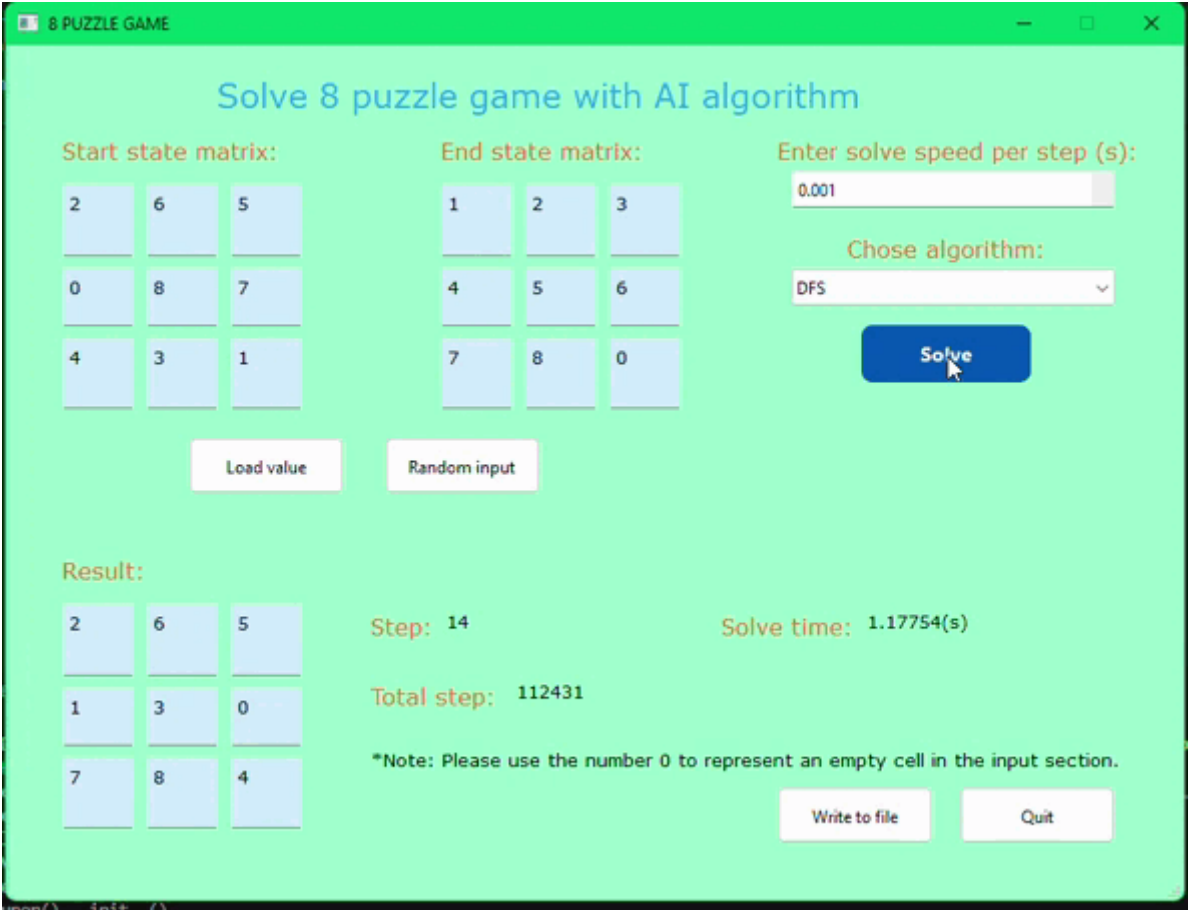
Tổng chi phí của đường đi từ trạng thái ban đầu đến trạng thái đang xét, thường ký hiệu là $g(\text{state})$ [2, tr. 168]

Giải pháp là một đường đi chứa các trạng thái với trạng thái đầu tiên là trạng thái ban đầu, các trạng thái biến đổi sau khi thực hiện hành động lên trạng thái ban đầu và trạng thái cuối cùng là trạng thái mục tiêu (trạng thái cần tìm).

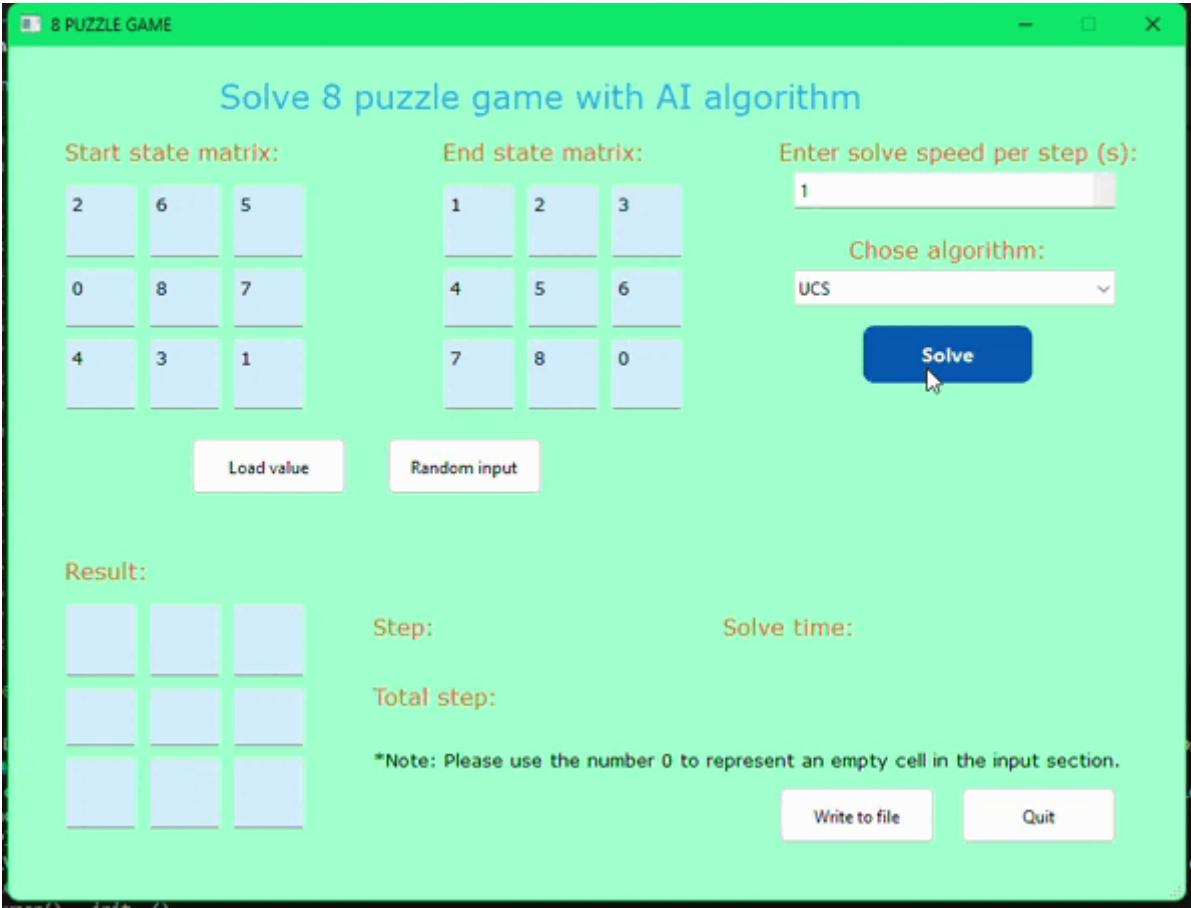
Thuật toán BFS



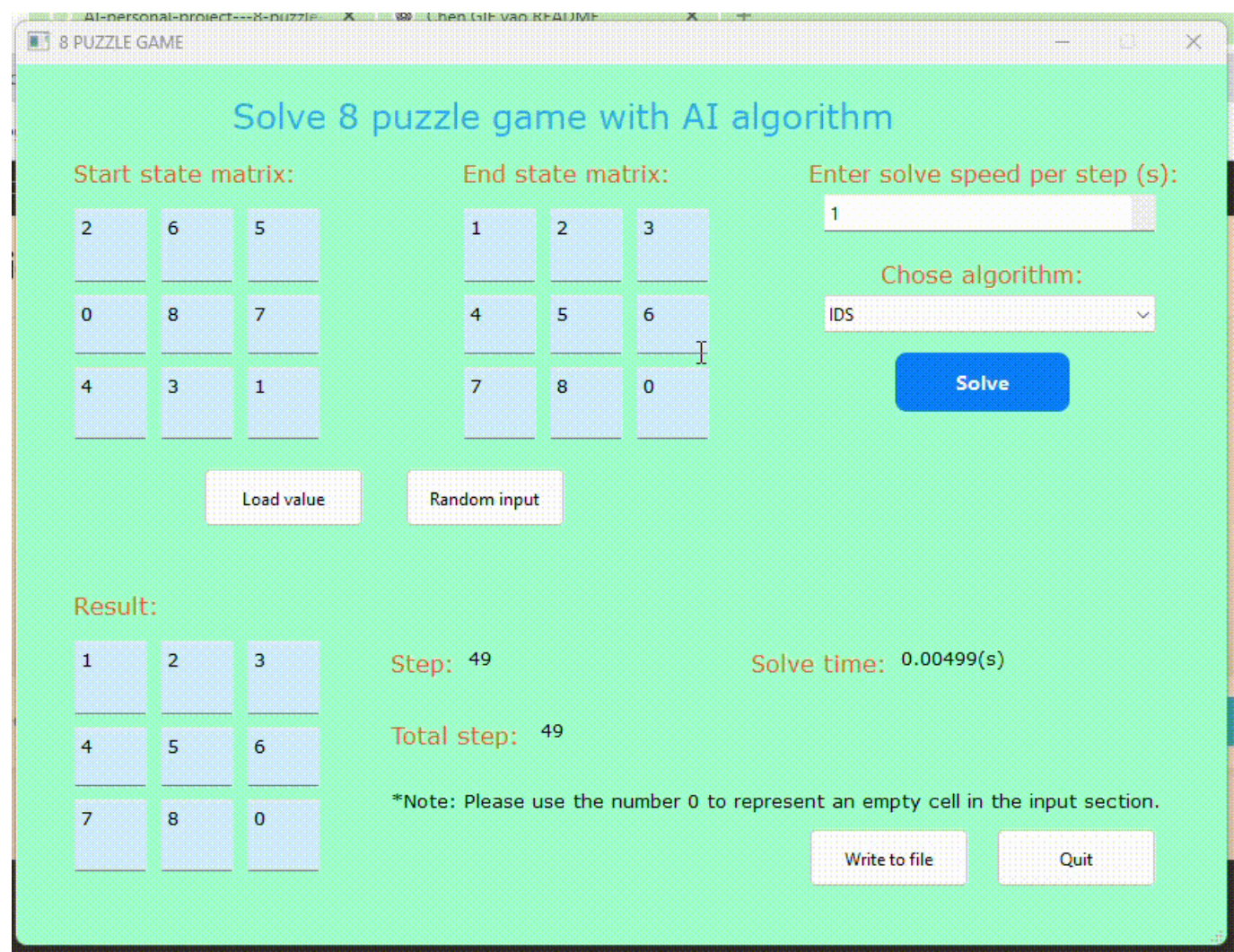
Thuật toán DFS



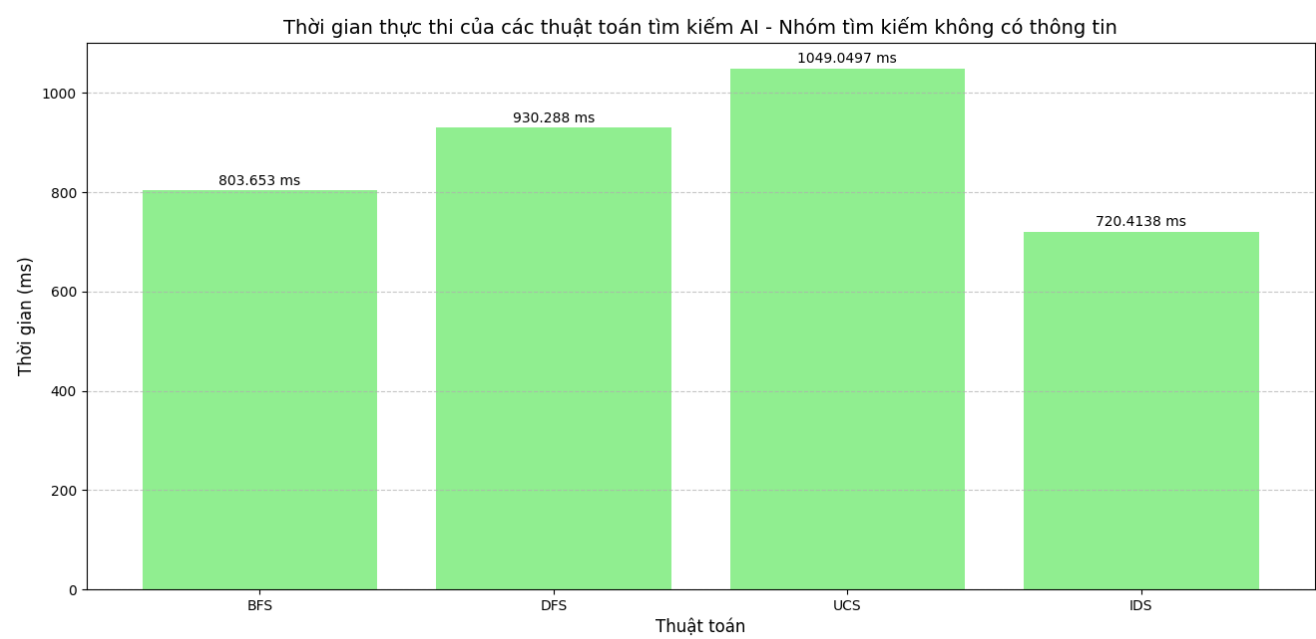
Thuật toán UCS - Uniform Cost Search



Thuật toán IDS



Hình ảnh so sánh hiệu suất của các thuật toán



Nhận xét về hiệu suất của các thuật toán trong nhóm

BFS - Breadth-First Search (Tìm kiếm theo chiều rộng): Khám phá các trạng thái theo mức độ, có thể được dùng khi các hành động có cùng chi phí, đảm bảo tìm được giải pháp ngắn nhất nếu tồn tại giải pháp. Thuật toán này dùng hàng đợi để lưu trữ các trạng thái đang được xét theo nguyên tắc FIFO (vào trước ra trước). Thuật toán kết thúc khi tìm ra lời giải hoặc khi hàng đợi rỗng. [2, tr.175]

DFS - Depth-First Search (Tìm kiếm theo chiều sâu): Khi các hành động có cùng chi phí có thể áp dụng thuật toán này để Khám phá sâu vào các nhánh trước khi quay lại xét nhánh kế tiếp. Thuật toán có thể tìm ra được lời giải nhưng không đảm bảo tìm được giải pháp tối ưu do có thể phải xét hết tất cả các trạng thái trên nhánh không có lời giải, sau đó mới chuyển sang các nhánh khác.

- UCS - Uniform Cost Search (Tìm kiếm theo chi phí đồng nhất): Mở rộng từ một trạng thái tới trạng thái có chi phí tốt nhất từ trạng thái gốc đến trạng thái đó, đảm bảo tìm được giải pháp tối ưu với chi phí hành trình tìm kiếm là thấp nhất do tính chi phí các nút lân cận và sau đó mới di chuyển đến nút có chi phí tốt nhất.

IDS - Iterative Deepening Search (Tìm kiếm sâu dần): Thuật toán này 'kết hợp ưu điểm của tìm kiếm theo chiều rộng và chiều sâu, thuật toán tìm kiếm theo chiều sâu từ mức thấp đến mức cao hơn, đến khi tìm ra giải pháp'. Đây là lựa chọn tối ưu với "các bài toán tìm kiếm khi không biết trước độ sâu của lời giải". [3]

3.2. Tìm kiếm có thông tin

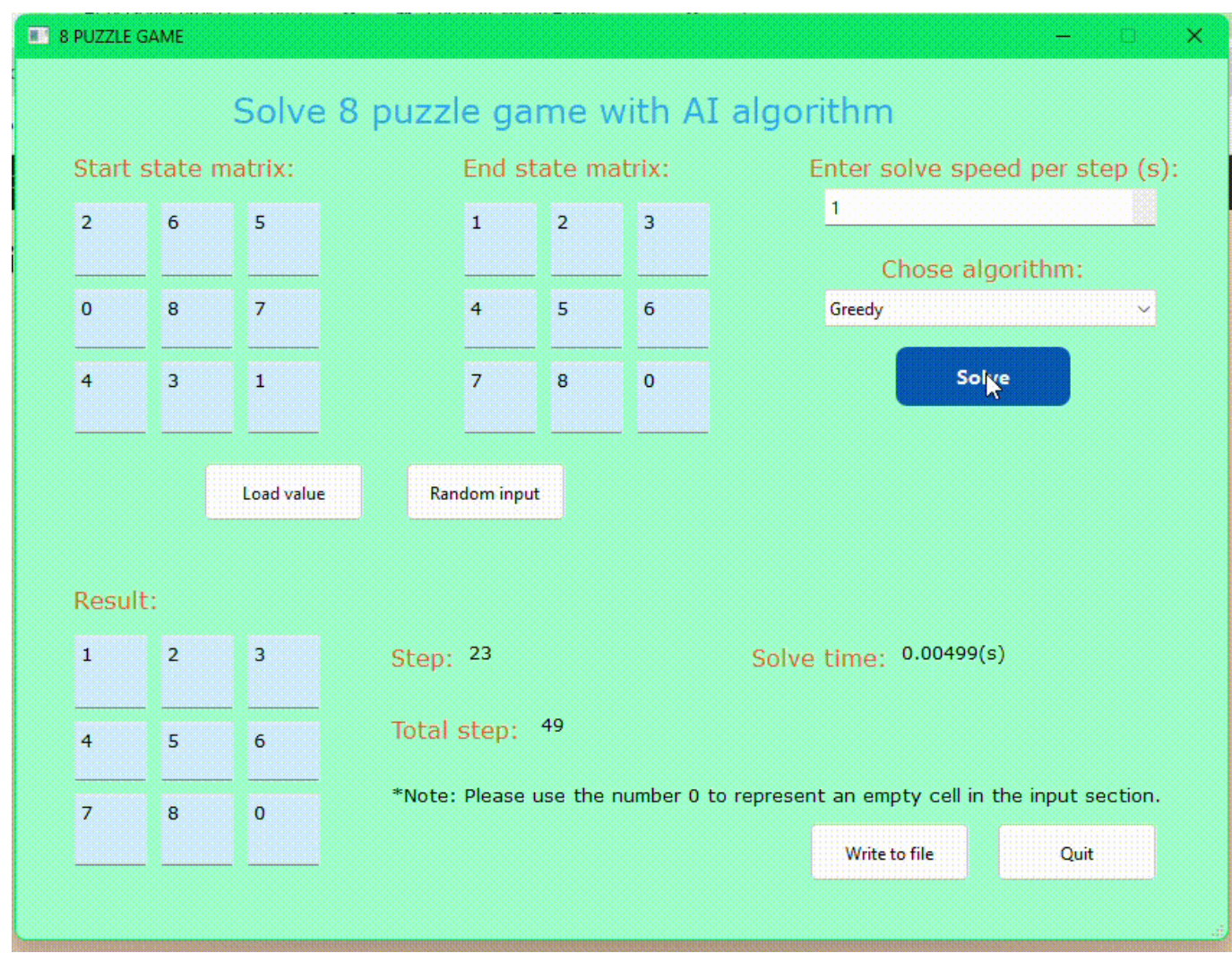
Các thành phần chính của bài toán tìm kiếm

Bài toán 8 puzzle có trạng thái ban đầu là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu vào bài toán cần giải quyết.

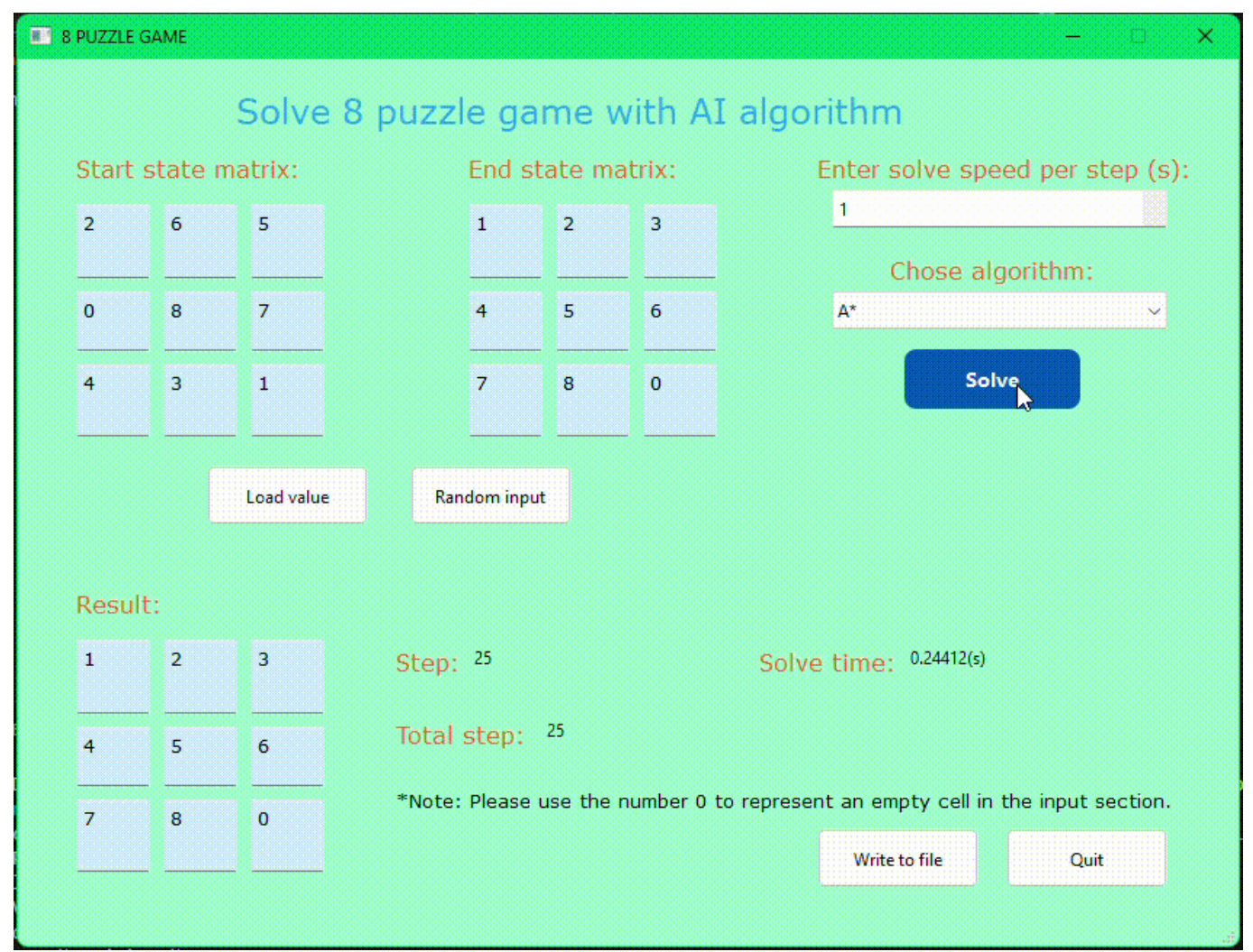
Bài toán 8 puzzle có trạng thái mục tiêu cũng là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu ra của bài toán (là trạng thái muốn có sao khi thực hiện các hành động từ trạng ban đầu).

Giải pháp là một đường đi chứa các trạng thái với trạng thái đầu tiên là trạng thái ban đầu, các trạng thái biến đổi sau khi thực hiện hành động lên trạng thái ban đầu và trạng thái cuối cùng là trạng thái mục tiêu (trạng thái cần tìm).

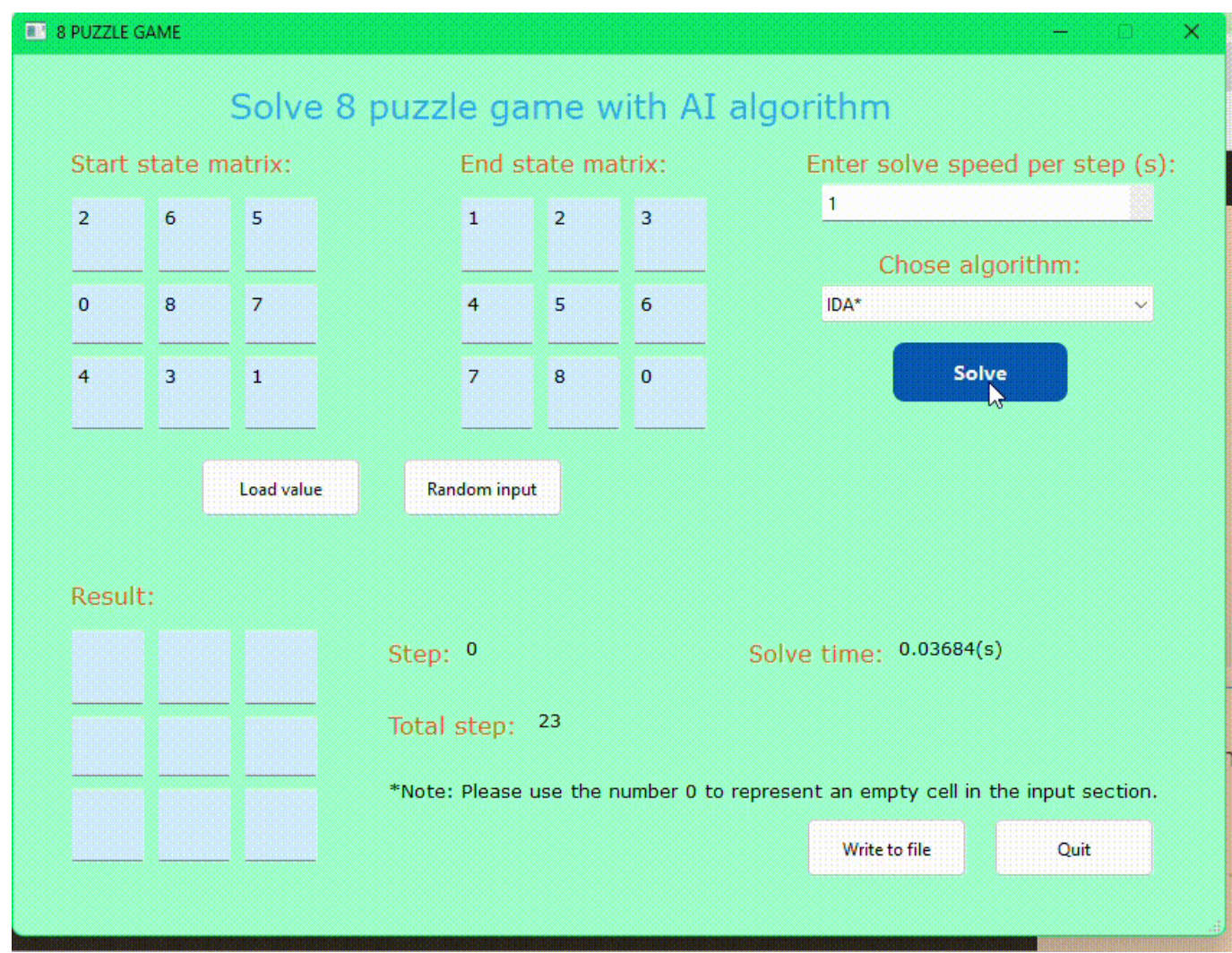
Thuật toán Greedy



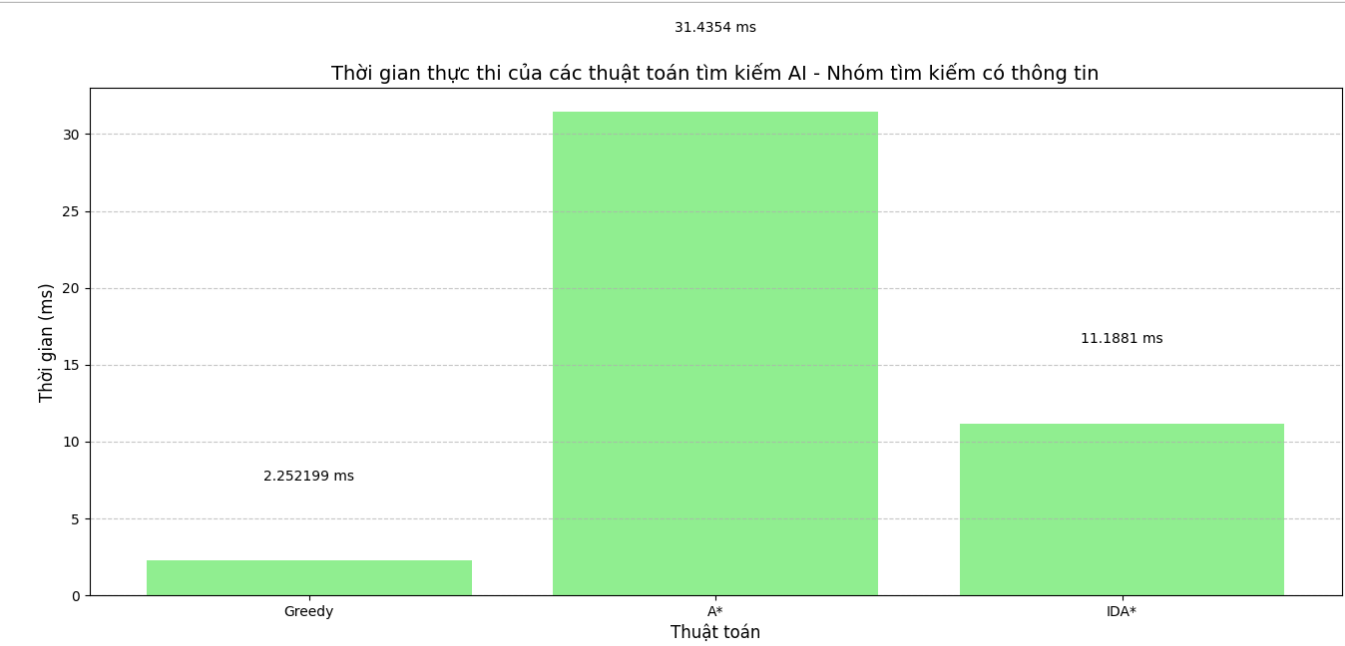
Thuật toán A*



Thuật toán Iterative deepening a*



Hình ảnh so sánh hiệu suất của các thuật toán



Nhận xét về hiệu suất của các thuật toán trong nhóm

Greedy Search (Tìm kiếm tham lam - tìm kiếm Greedy): Mở rộng trạng thái tới trạng thái có giá trị hàm heuristic tốt nhất (trong đó, hàm heuristic là hàm đánh giá chi phí từ trạng thái đang xét đến trạng thái đích).

A-Star Search (Tìm kiếm A*): Tìm kiếm bằng cách tính chi phí từ trạng thái ban đầu đến trạng thái hiện tại và ước lượng chi phí từ trạng thái hiện tại đến trạng thái mục tiêu để tìm ra trạng thái tiếp theo có chi phí tốt nhất để di chuyển đến.

IDA-Star ...

3.3. Thuật toán tìm kiếm cục bộ

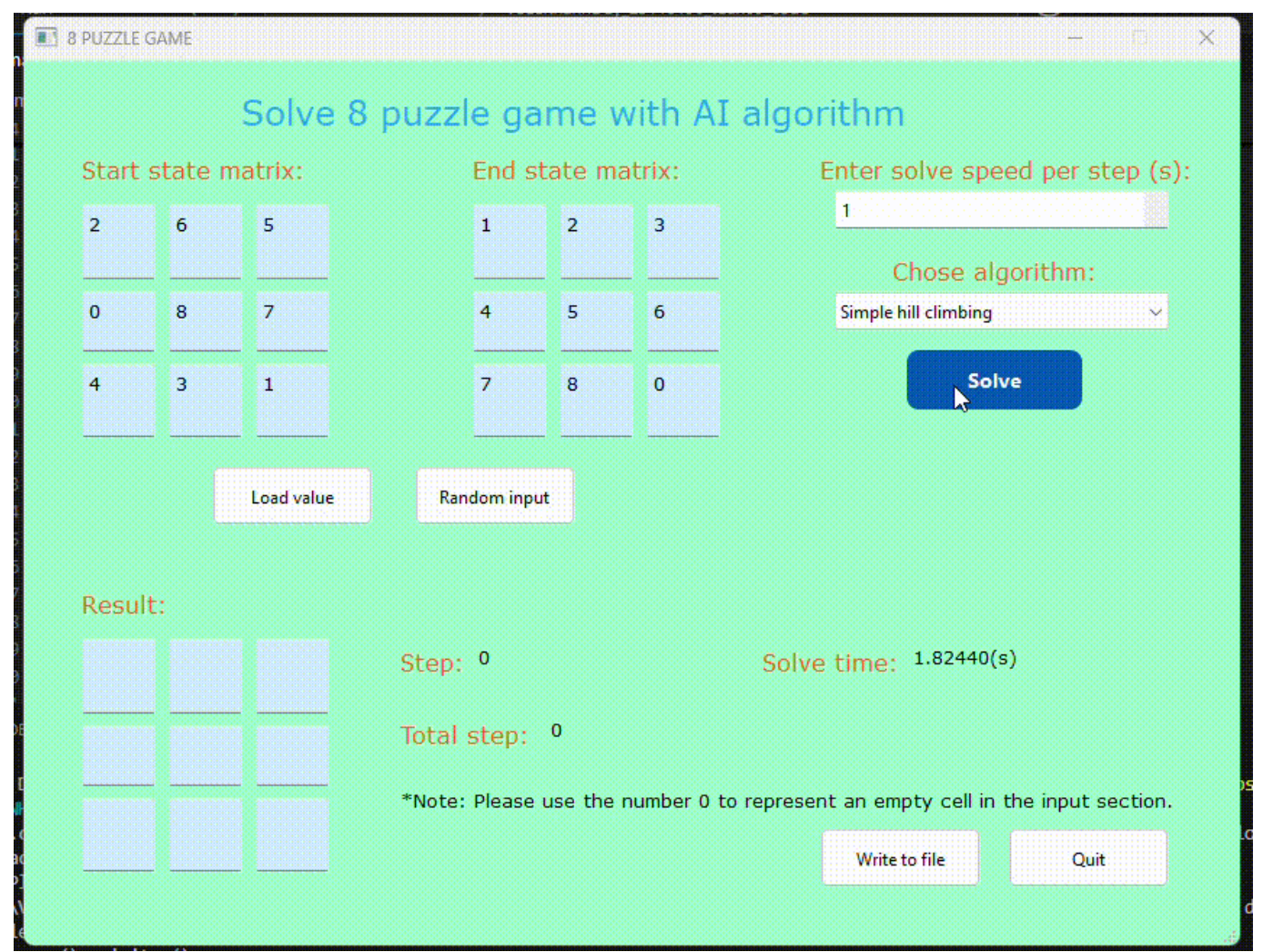
Các thành phần chính của bài toán tìm kiếm

Bài toán 8 puzzle có trạng thái ban đầu là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu vào bài toán cần giải quyết.

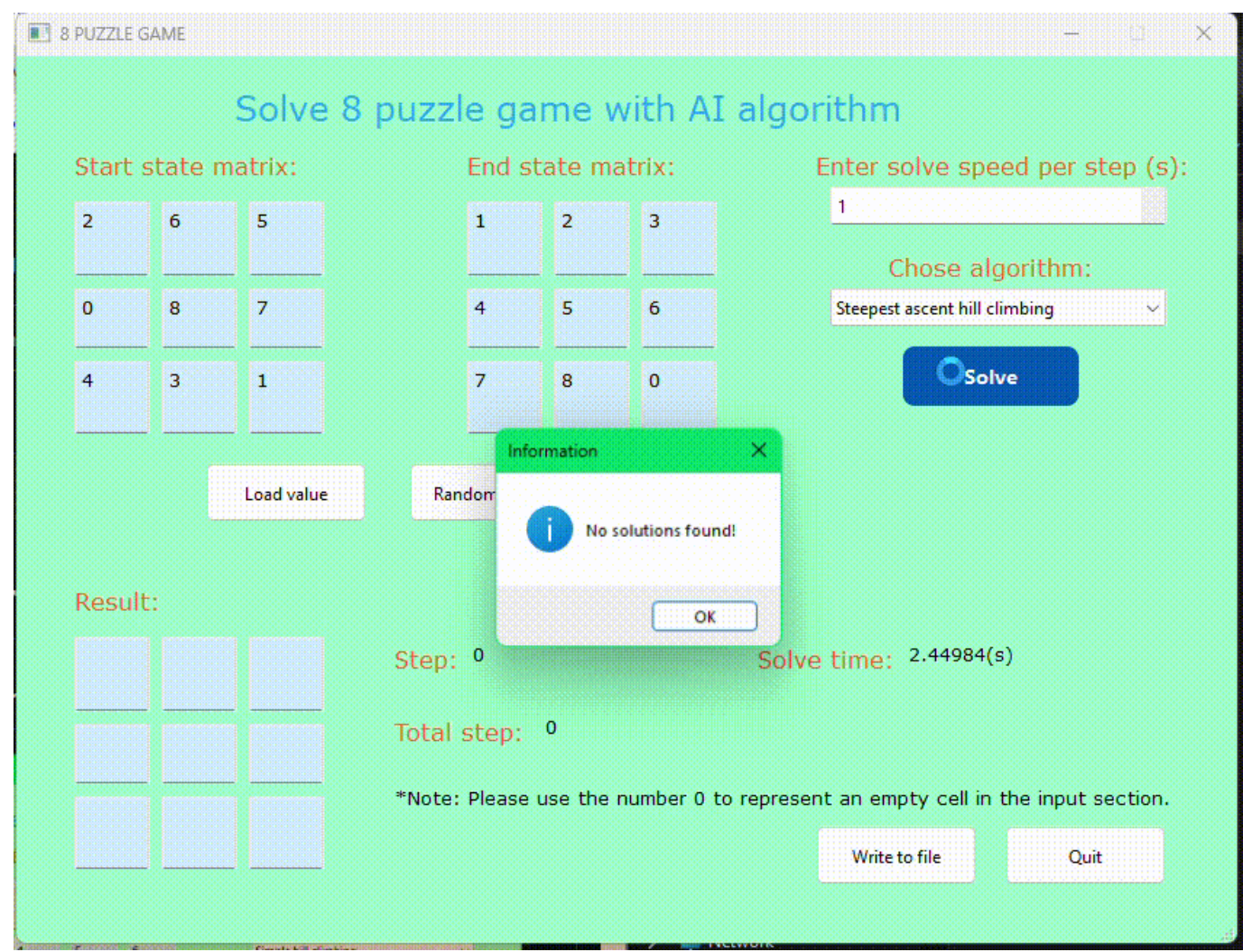
Bài toán 8 puzzle có trạng thái mục tiêu cũng là ma trận 3×3 chứa 9 chữ số từ 0 đến 8 không trùng lặp (với số 0 đại diện cho ô trống) là đầu ra của bài toán (là trạng thái muốn có sau khi thực hiện các hành động từ trạng thái ban đầu).

Giải pháp là một đường đi chứa các trạng thái với trạng thái đầu tiên là trạng thái ban đầu, các trạng thái biến đổi sau khi thực hiện hành động lên trạng thái ban đầu và trạng thái cuối cùng là trạng thái mục tiêu (trạng thái cần tìm).

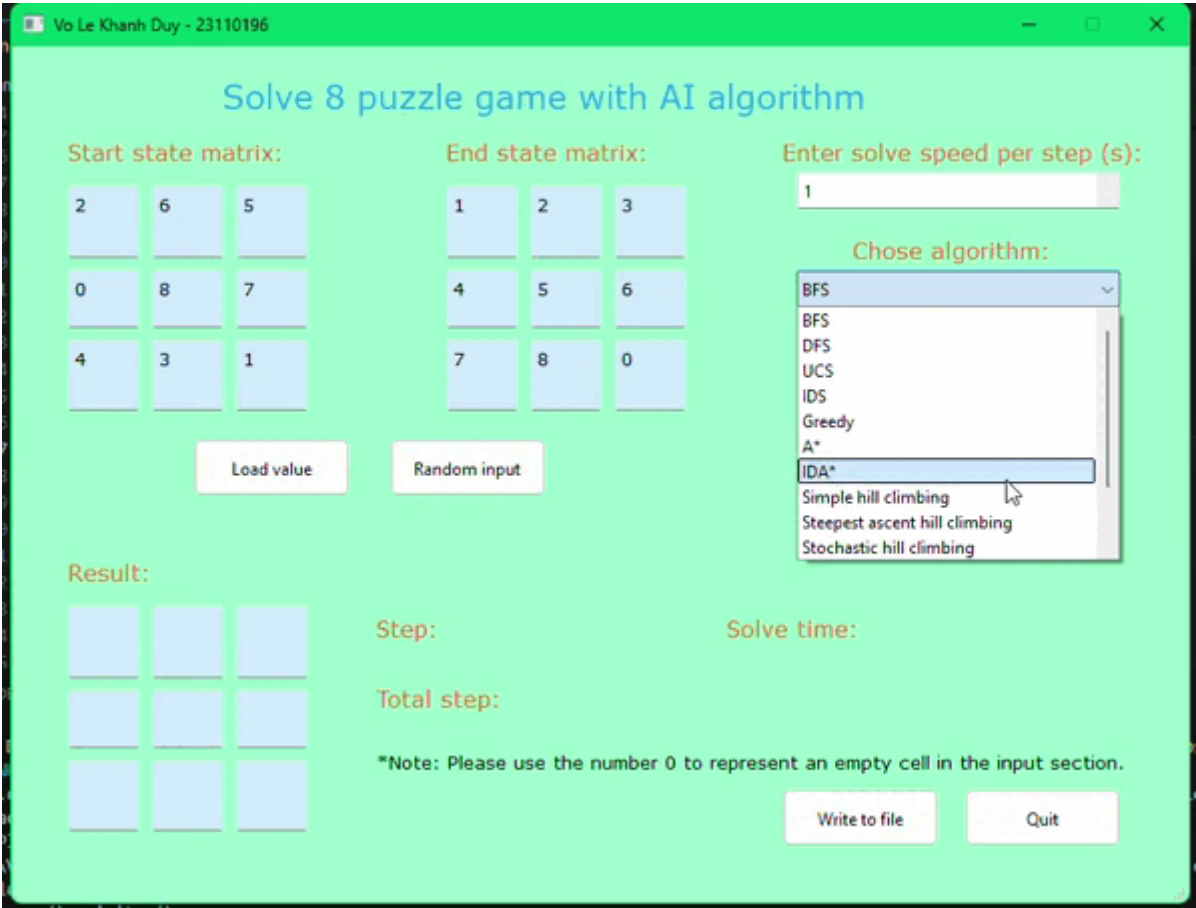
Thuật toán Simple hill climbing



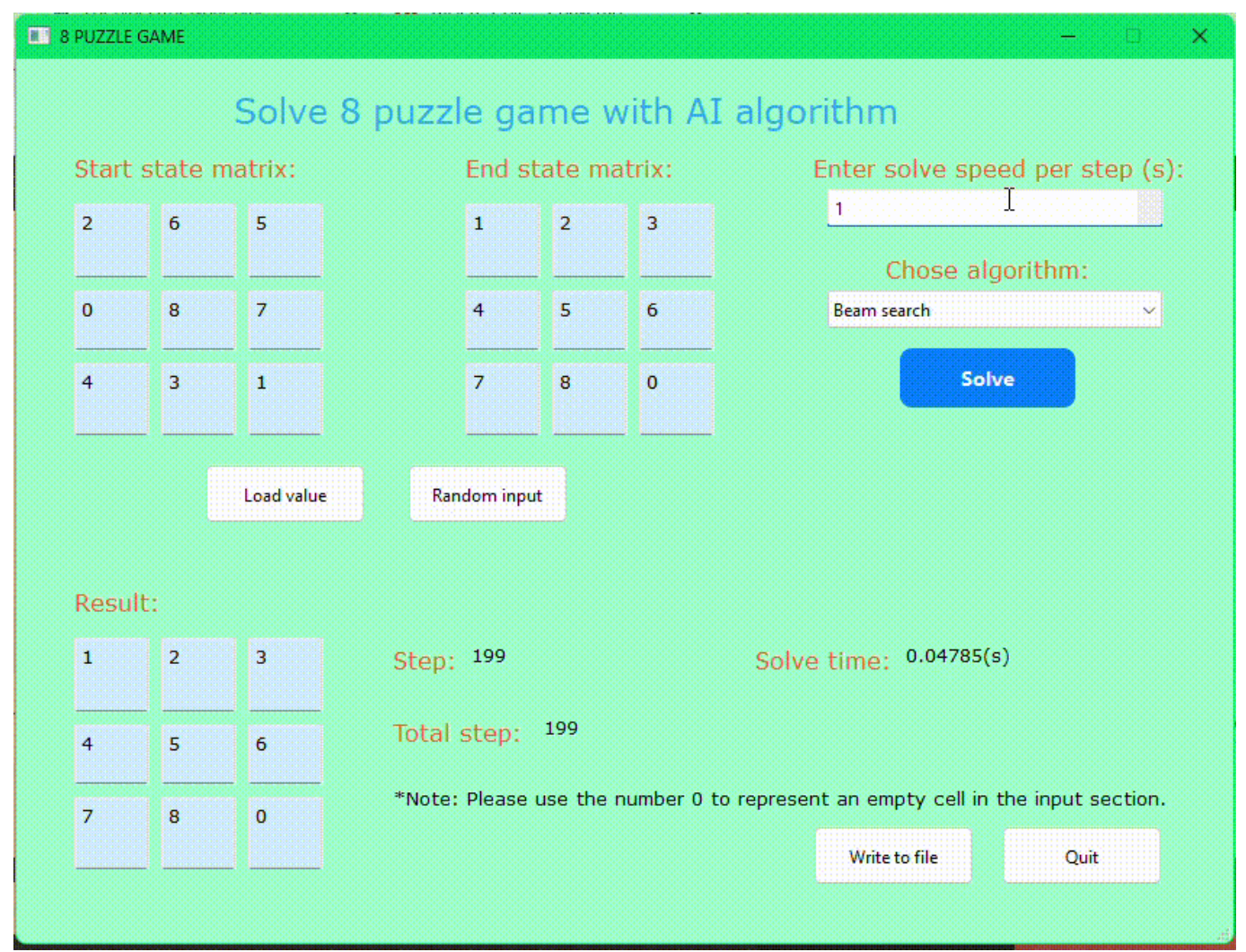
Thuật toán Stepppest ascent hill climbing



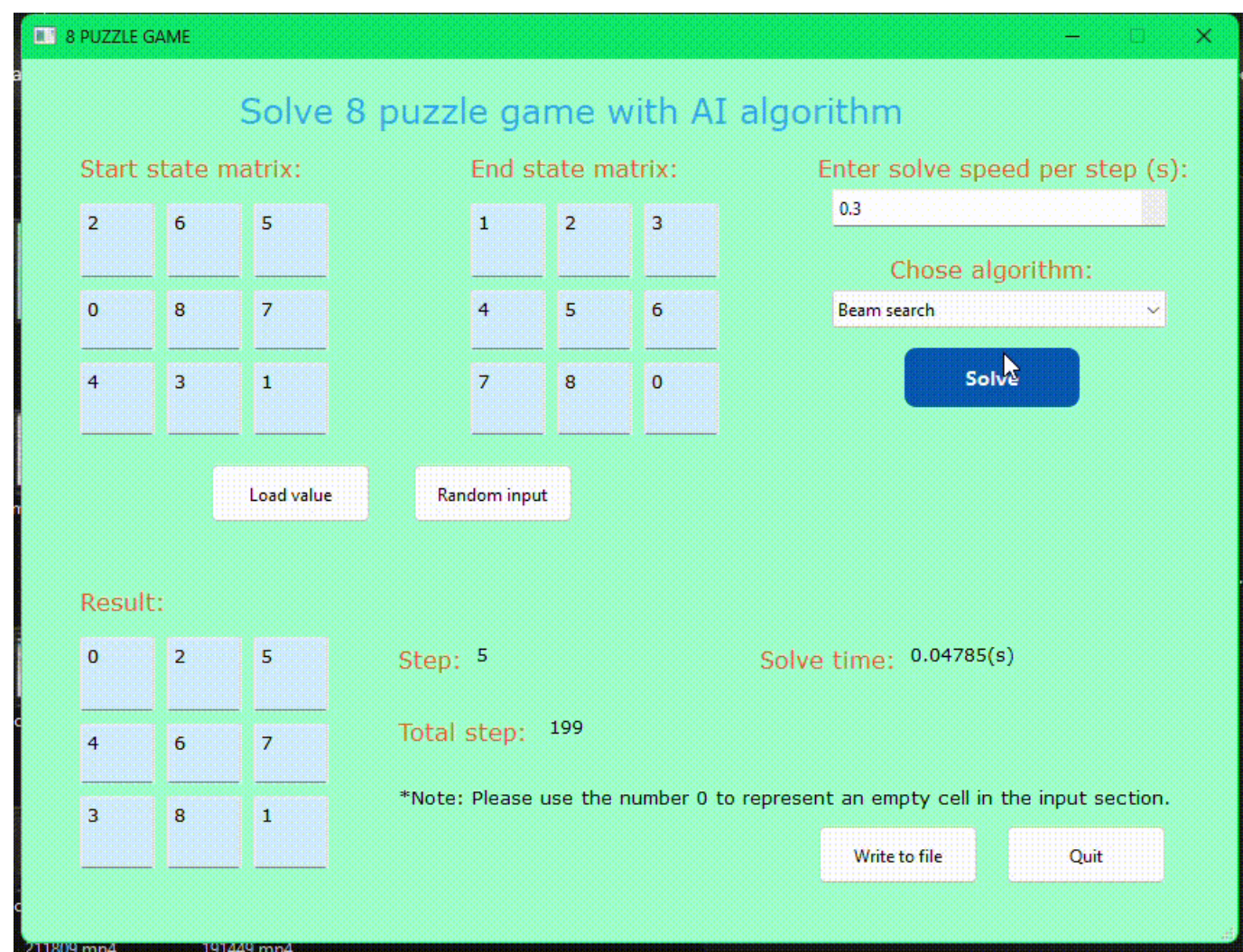
Thuật toán Stochastic hill climbing



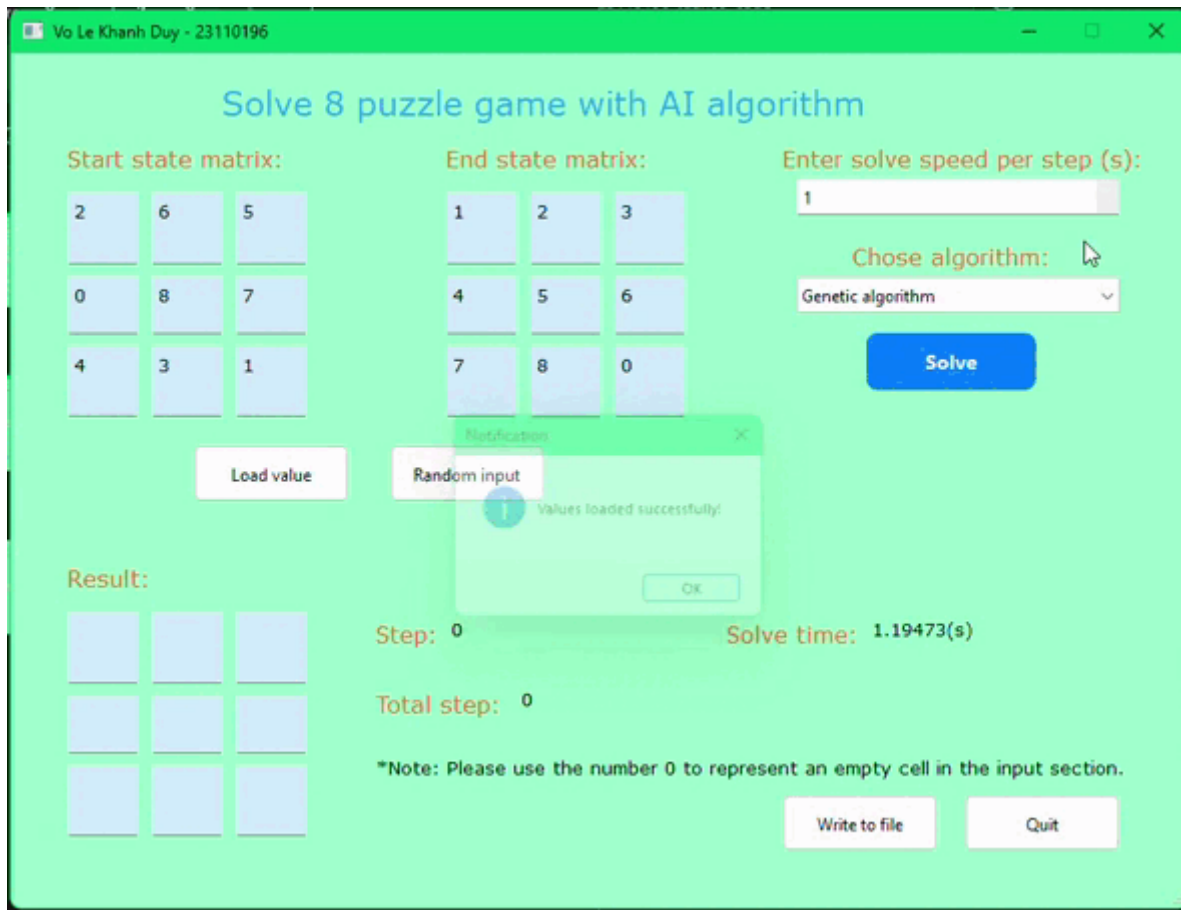
Thuật toán Stimulated Annealing



Thuật toán Beam Search



Thuật toán Genetic Algorithm



Hình ảnh so sánh hiệu suất của các thuật toán

Nhận xét về hiệu suất của các thuật toán trong nhóm

Nhìn chung ở nhóm thuật toán này, từ một trạng thái hiện tại đánh xét, quá trình tìm kiếm sẽ mở rộng ra các trạng thái lân cận tốt hơn mà không xét hết cây tìm kiếm. Nhược điểm lớn nhất ở nhóm này là quá trình tìm kiếm dễ mắc kẹt tại cực trị địa phương dẫn đến không tìm ra lời giải.

Thuật toán leo đồi đơn giản đánh giá từng trạng thái lân cận một cách tuần tự và chọn trạng thái đầu tiên tối ưu hơn so với trạng thái hiện tại.

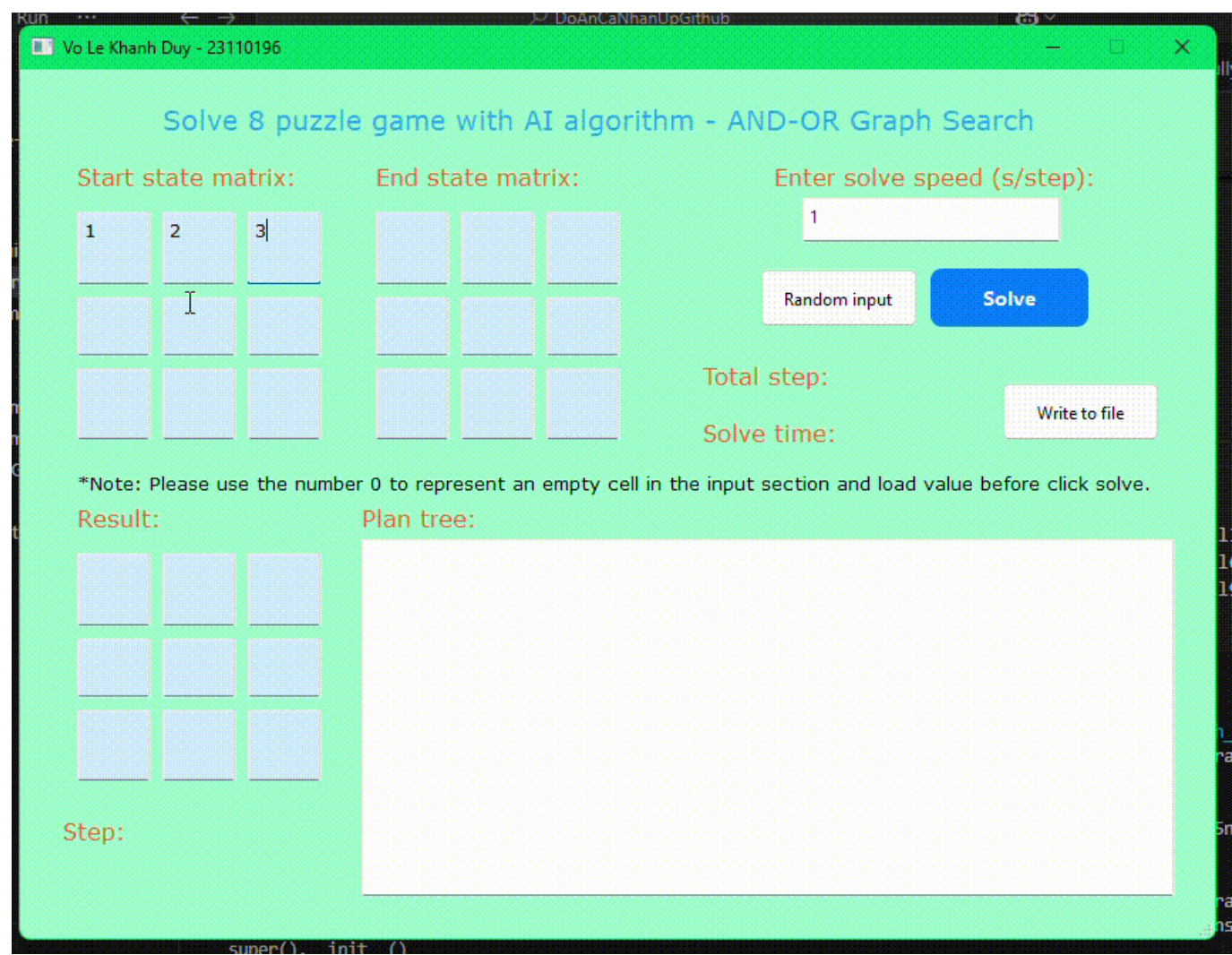
Thuật toán leo đồi dốc nhất đánh giá tất cả các trạng thái lân cận và chọn trạng thái mang lại cải thiện lớn nhất so với trạng thái hiện tại.

Thuật toán leo đồi ngẫu nhiên chọn ngẫu nhiên một trạng thái lân cận và quyết định chuyển sang trạng thái đó nếu nó tốt hơn trạng thái hiện tại.

3.3. Thuật toán tìm kiếm trong môi trường phức tạp

Các thành phần chính của bài toán tìm kiếm

Thuật toán tìm kiếm AND - OR graph search



Thuật toán tìm kiếm không có sự quan sát

(search with no observation)

Solve 8 puzzle game with AI algorithm - Search in complex environment

Start state matrix:

1	2	3
4	5	6
0	7	8

End state matrix:

1	2	3
4	5	6
7	8	0

1	2	3
4	0	6
7	5	8

Enter solve speed (s/step):

1

Chose algorithm name:

Search with no observation

Random input

Solve

Step: 3

Total step: 3

Solve time: 0.0708928000(s)

Result:

1	2	3
4	5	6
7	8	

1	2	3
4	5	6
7	8	

1	2	3
4	5	6
7	8	

1	2	3
4	5	6
7	8	

*Note: Please use the number 0 to represent an empty cell in the input section.

Write to file

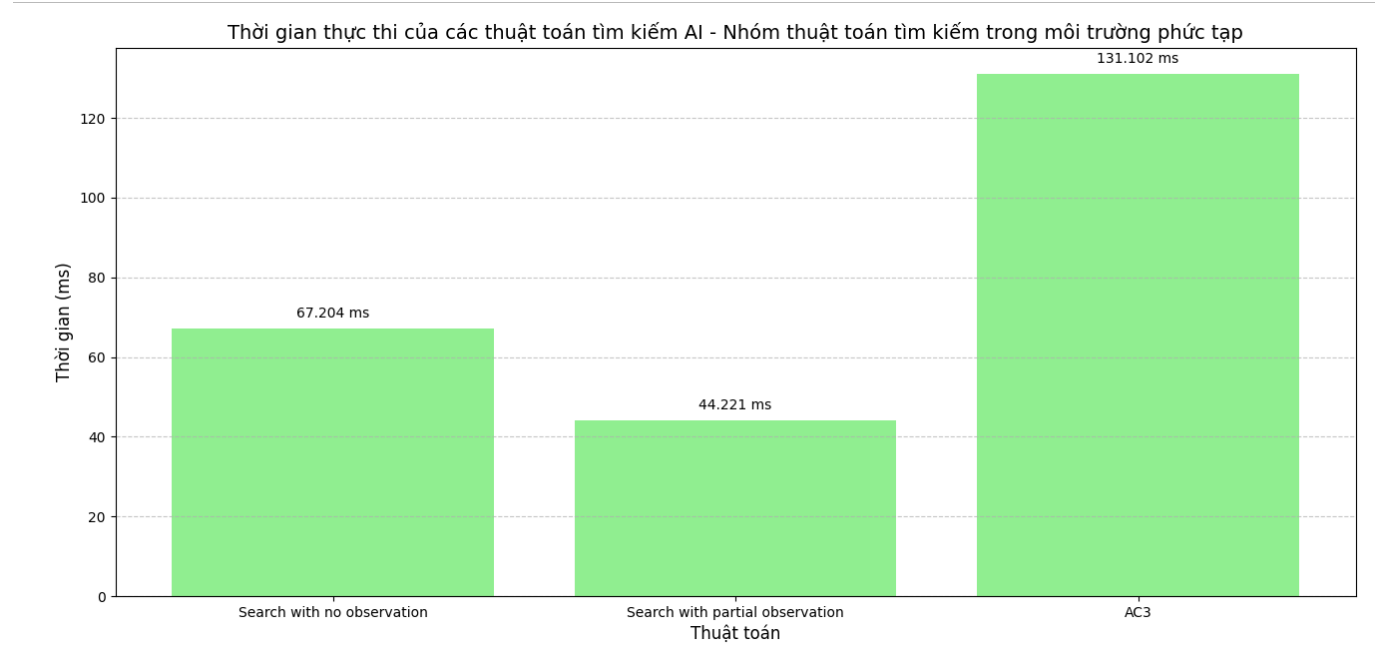
Thuật toán tìm kiếm có sự quan sát một phần

(search with partial observation)

19 / 25



Hình ảnh so sánh hiệu suất của các thuật toán



3.4. Thuật toán tìm kiếm có ràng buộc

Các thành phần chính của bài toán tìm kiếm

Thuật toán kiểm thử

Vo Le Khanh Duy - 23110196

Solve 8 puzzle game with AI algorithm - Constrain satisfaction problem

End state matrix:

X1X2X3

X4X5X6

X7X8X9

Input domain of variables:

MINMAX

X108

X208

X308

X408

X508

X608

X708

X808

X908

Enter solve speed (s/step):

0.01

Enter algorithm name:

Test

Solve

Result:

660

156

356

Step: 20

Total step: 357

Solve time: 0.2807126000(s)

*Note: Please use the number 0 to represent an empty cell in the input section.

Write to file

Output domain of variables:

X1: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X2: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X3: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X4: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X5: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X6: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X7: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X8: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X9: [0, 1, 2, 3, 4, 5, 6, 7, 8]

Thuật toán backtracking

21 / 25

Võ Lê Khanh Duy - 23110196

Solve 8 puzzle game with AI algorithm - Constrain satisfaction problem

End state matrix:

X1

X2

X3

X4

X5

X6

X7

X8

X9

Input domain of variables:

MIN

MAX

X1

1

1

X2

0

8

X3

0

8

X4

0

8

X5

3

3

X6

0

8

X7

0

8

X8

0

8

X9

0

8

Enter solve speed (s/step):

1

Enter algorithm name:

Backtracking

Solve

Result:

1

0

2

4

3

5

6

7

8

Step: 11

Total step: 11

Solve time: 0.0448169000(s)

*Note: Please use the number 0 to represent an empty cell in the input section.

Output domain of variables:

X1: [1]

X2: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X3: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X4: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X5: [3]

X6: [0, 1, 2, 3, 4, 5, 6, 7, 8]

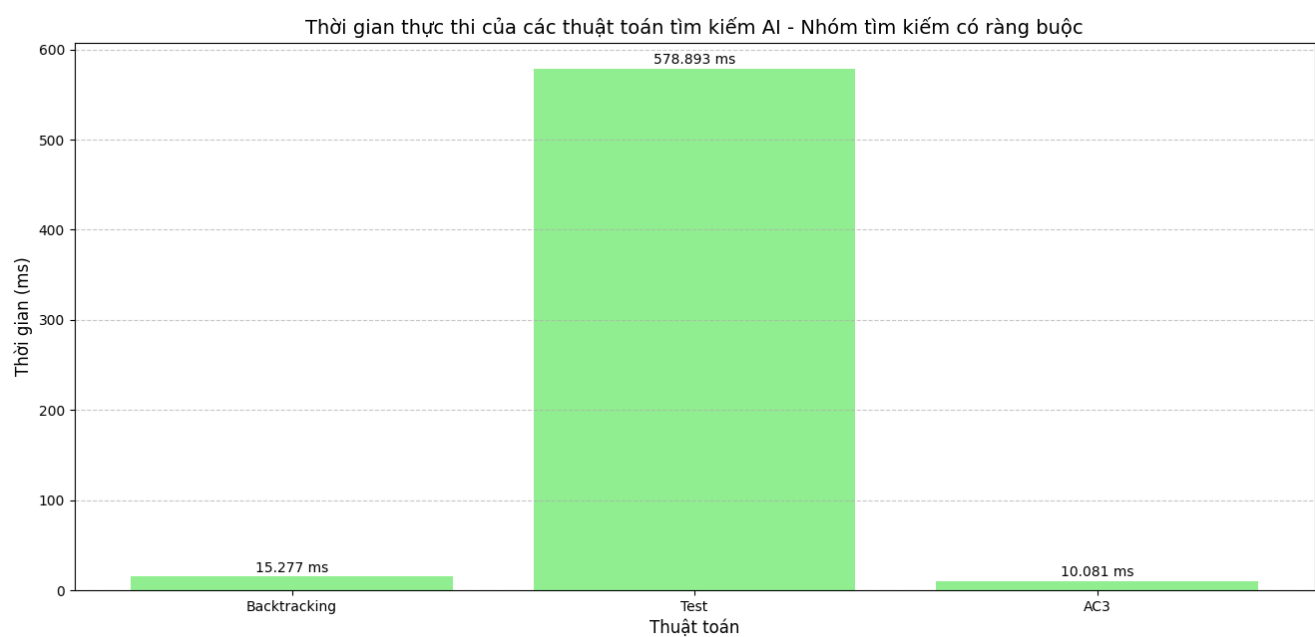
X7: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X8: [0, 1, 2, 3, 4, 5, 6, 7, 8]

X9: [0, 1, 2, 3, 4, 5, 6, 7, 8]

Write to file

Hình ảnh so sánh hiệu suất của các thuật toán



Nhận xét về hiệu suất của các thuật toán trong nhóm

3.5. Thuật toán tìm kiếm học tăng cường

Các thành phần chính của bài toán tìm kiếm

Thuật toán tìm kiếm Q Learning

Võ Lê Khanh Duy - 23110196

Solve 8 puzzle game with AI algorithm

Start state matrix:

1

2

3

4

5

6

7

8

0

End state matrix:

0

1

2

4

5

3

7

8

6

Enter solve speed per step (s):

1

Chose algorithm:

BFS

Solve

Random input

Result:

Step:

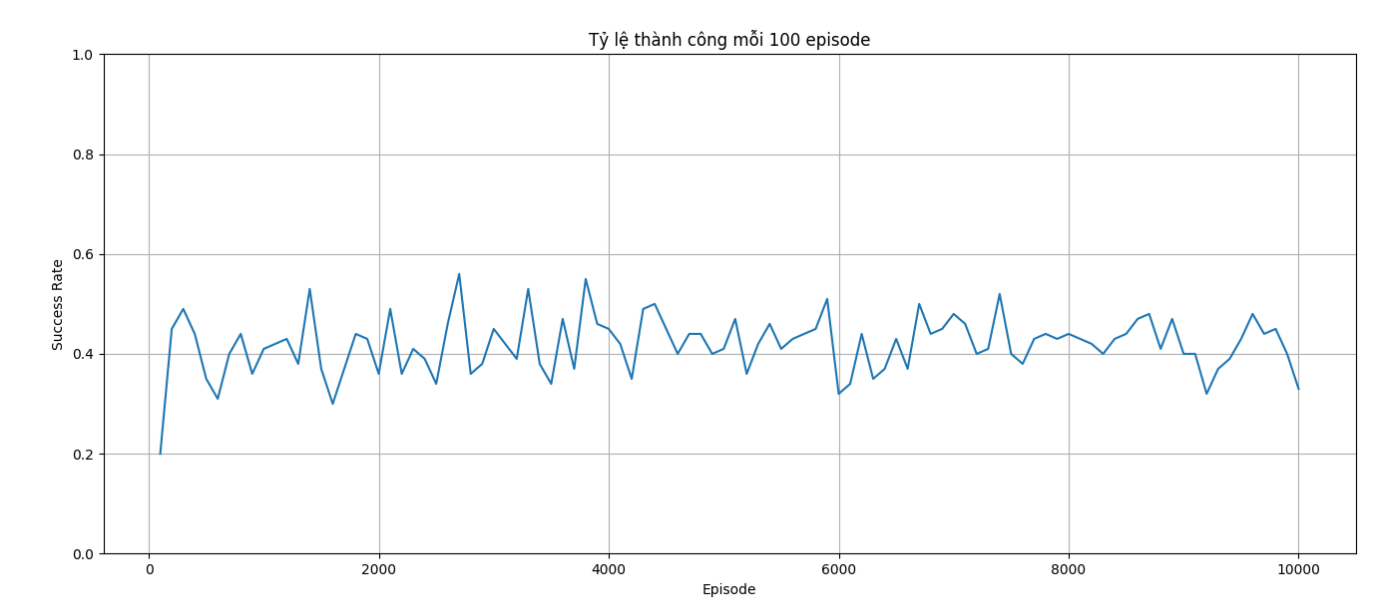
Solve time:

Total step:

*Note: Please use the number 0 to represent an empty cell in the input section.

Write to file

Hình ảnh so sánh hiệu suất của các thuật toán





Nhận xét về hiệu suất của thuật toán

4. Kết luận

5. Tài liệu tham khảo

- [1]. Elastic Platform Team, "Understanding AI search algorithms", elastic, <https://www.elastic.co/blog/understanding-ai-search-algorithms>, ngày 21 tháng 3 năm 2024 (truy cập ngày 9 tháng 5 năm 2025)
- [2]. Stuart Russell and Peter Norvig, "Russell 2020 Artificial intelligence a modern approach", xuất bản lần 4
- [3]. "Các thuật toán tìm kiếm: chìa khóa mở cửa trí tuệ nhân tạo", KDATA, <https://kdata.vn/tin-tuc/cac-thuat-toan-tim-kiem-chia-khoa-mo-cua-tri-tue-nhan-tao>, (truy cập ngày 6 tháng 5 năm 2025)