

MSSV: 23520442

Họ tên: Phạm Thị Hồng Hạnh

Lớp: CS112.P22

BÀI TẬP BONUS

THIẾT KẾ PHẦN MỀM GIẢI BÀI TOÁN MÃ ĐI TUẦN

- **Phát biểu bài toán:**

- Mục tiêu: Đặt quân mã tại 1 ô bất kỳ, thiết kế giải thuật sao cho quân mã có thể đi qua mọi ô trên bàn cờ (theo đúng luật cờ vua) với điều kiện mỗi ô chỉ được đi qua 1 lần.
- Input: 2 số nguyên $startX$ và $startY$ là vị trí xuất phát theo dòng và theo cột. Với $0 \leq startX, startY \leq 7$
- Output: Danh sách các vị trí quân mã đi qua theo đúng thứ tự .

- **Các chức năng của phần mềm:**

- Giải bài toán mã đi tuần với vị trí xuất phát bất kỳ trên bàn cờ 8×8 .
- Cho phép người dùng chọn $startX$, $startY$ là tọa độ muốn đặt quân mã trên bàn cờ để bắt đầu bài toán.
- Giải bài toán mã đi tuần với vị trí xuất phát là vị trí người dùng nhập sau đó lời giải sẽ hiện lên màn hình với mỗi ô là thứ tự của ô trên đường di chuyển của quân mã để có thể đi khắp các ô trên bàn cờ mà không lặp lại. Bắt đầu từ 0 là vị trí xuất phát và 63 là vị trí kết thúc.
- Cung cấp các chức năng tạm dừng và tiếp tục quá trình hiển thị kết quả, tạo điều kiện cho người dùng dễ dàng quan sát đường đi của quân mã.
- Có thể hiển thị các bước giải. Trong đó 1 ô khi được tô màu xanh lá là ô đã được chọn tại bước hiện tại với nội dung là bước hiện tại, với ô này sẽ có các ô có thể tới tại bước tiếp theo, những ô này được tô màu xanh dương. Các ô có thể đến này sẽ hiển thị số ô có thể đi đến tại bước tiếp theo của ô đó. Trong số những ô này, ô nào có số ô có thể đi đến nhỏ nhất sẽ được tô màu hồng là ô tiếp theo được chọn. Giả sử quá trình thử của lời giải tại bước hiện tại thất bại, màu của ô sẽ trở về ban đầu và nội dung lưu trong ô sẽ biến mất.

- **Ý tưởng**

- Phương pháp chính: Kết hợp giữa backtracking và thuật toán tham lam Warnsdorff.
- Nếu chỉ sử dụng backtracking: độ phức tạp lớn, lên tới $O(8^{N^2})$ với N là kích thước bàn cờ, sẽ khiến người dùng phải đợi mãi cho đến khi đạt được kết quả mong muốn.
- Thuật toán Warnsdorff là 1 thuật toán tham lam, tại đó thay vì duyệt qua hết các ô có thể đi tới từ vị trí hiện tại theo thứ tự index, thì ưu tiên đi tới các ô mà có các nước đi tiếp theo ít hơn từ đó giảm thiểu số phép thử và quay lui nếu như ô đó không đưa ta đến hết bàn cờ.
- Tuy nhiên thuật toán em đang sử dụng không hoàn toàn giống Warnsdorff mà chỉ kết hợp tư duy ở việc ưu tiên các ô có ít nước đi hợp lệ kế tiếp nhất và kết hợp với backtracking để đảm bảo luôn tìm được kết quả đúng.
- Độ phức tạp của phương pháp này là $O(8^{N^2})$ trong trường hợp tệ nhất khi mà hướng đi đó không đúng. Tuy nhiên trung bình thì nó sẽ có độ phức tạp là $O(N^2)$ do việc lựa chọn ô tiếp theo theo chiến thuật này thường rất ít khi cần phải quay lui và luôn tìm được kết quả khi chỉ đi theo nhánh đó.
- **Chi tiết thuật toán**
 - Tại mỗi bước, từ vị trí hiện tại (x, y), duyệt 8 hướng quân mã có thể đi.
 - Với mỗi ô hợp lệ (newX, newY), đếm số lượng ô mà quân mã có thể tiếp tục đi tiếp từ ô đó — gọi là onwardCount.
 - Tạo danh sách nextMoves chứa các ô (newX, newY, onwardCount) và sắp xếp tăng dần theo onwardCount.
 - Thử đi đến từng ô theo thứ tự đã sắp xếp, đệ quy gọi lại SolveKnightTour cho từng ô với step+1.
 - Nếu tìm được lời giải return True. Ngược lại, thực hiện quay lui với thao tác gỡ bước đó ra khỏi path và đặt lại board[newX, newY] = -1.
- **Cấu trúc dữ liệu:**
 - Ma trận vuông cỡ NxN (N=8) để thứ tự “ghé thăm” của quân mã. Ban đầu, các ô sẽ mang giá trị là -1, thể hiện là quân mã chưa đi tới ô này. Sau khi nhấn nút Solve, quá trình giải bài toán sẽ bắt đầu. Khi kết thúc và đưa ra lời giải, ô ở vị trí xuất phát sẽ mang giá trị là 0, ô ở vị trí kết thúc mang giá trị là 63.
 - Danh sách List<(N,N)> path với path[i] là ô được duyệt qua tại bước i.
 - 2 Mảng 1 chiều kích thước N là dx, dy lưu bước di chuyển theo luật của quân mã. Quá trình kiểm tra tính hợp lệ của nước đi sẽ đảm bảo quân mã chỉ di chuyển trong bàn cờ không vào lại các ô đã đi qua.

- Với mỗi bước di chuyển hiện tại, có 1 List<(int nextX, int nextY, int onwardCount)> lưu danh sách các vị trí có thể đi đến từ bước hiện tại và onwardCount là biến lưu số bước hợp lệ tại vị trí (nextX, nextY).
- Link github đến project:
<https://github.com/HongHanh-atUIT/Knight-s-Tour.git>