# 1.Tìm kiếm theo chiều sâu DFS

*Quy tắc*

Qui tắc 1: Duyệt tiếp tới đỉnh liền kề mà chưa được duyệt. Đánh dấu đỉnh mà đã được duyệt. Hiển thị đỉnh đó và đẩy vào trong một ngăn xếp (stack).

Qui tắc 2: Nếu không tìm thấy đỉnh liền kề, thì lấy một đỉnh từ trong ngăn xếp (thao tác pop up). (Giải thuật sẽ lấy tất cả các đỉnh từ trong ngăn xếp mà không có các đỉnh liền kề nào)

Qui tắc 3: Lặp lại các qui tắc 1 và qui tắc 2 cho tới khi ngăn xếp là trống.

*Ưu điểm, nhược điểm*

Ưu điểm:

Xét duyệt tất cả các đỉnhđể trả về kết quả.

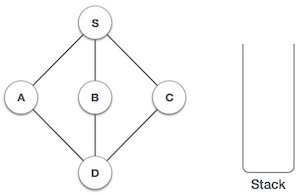
Nếu số đỉnh là hữu hạn, thuật toán chắc chắn tìm ra kết quả

Nhược điểm:

Mang tính chất vét cạn, không nên áp dụng nếu duyệt số đỉnh quá lớn.

Mang tính chất mù quáng, duyệt tất cả đỉnh, không chú ý đến thông tintrong các đỉnh để duyệt hiệu quả, dẫn đến duyệt qua các đỉnh không cầnthiết

*Ví dụ*



B1: Khởi tạo ngăn xếp (stack)\

B2: Đánh dấu đỉnh S là đã duyệt và đặt đỉnh này vào trong ngăn xếp. Tìm kiếm bất kỳ đỉnh liền kề nào mà chưa được duyệt từ đỉnh S. Chúng ta có 3 đỉnh và chúng ta có thể lấy bất kỳ đỉnh nào trong số chúng. Ví dụ, chúng ta lấy đỉnh A theo thứ tự chữ cái.

B3: Đánh dấu đỉnh A là đã duyệt và đặt vào trong ngăn xếp. Tìm kiếm bất kỳ đỉnh liền kề nào với đỉnh A. Cả S và D đều là hai đỉnh liền kề A nhưng chúng ta chỉ quan tâm về đỉnh chưa được duyệt.

B4: Duyệt đỉnh D, đánh dấu đỉnh này là đã duyệt và đặt vào trong ngăn xếp. Ở đây, chúng ta có B và C là hai đỉnh kề với D và cả hai đều là chưa được duyệt. Chúng ta sẽ chọn theo thứ tự chữ cái một lần nữa

B5: Chọn B, đánh dấu là đã duyệt và đặt vào trong ngăn xếp. Ở đây B không có bất kỳ đỉnh liền kề nào mà chưa được duyệt. Vì thế chúng ta lấy B ra khỏi ngăn xếp.

B6: Kiểm tra phần tử trên cùng của ngăn xếp để trở về nút đã duyệt trước đó và kiểm tra xem đỉnh này có đỉnh nào liền kề mà chưa được duyệt hay không. Ở đây, chúng ta tìm thấy đỉnh D nằm ở trên cùng của ngăn xếp.

B7: Chỉ có một đỉnh liền kề với D mà chưa được duyệt, đó là đỉnh C. Chúng ta duyệt C, đánh dấu là đã duyệt và đặt vào trong ngăn xếp.

Vì C không có bất kỳ đỉnh nào liền kề mà chưa được duyệt, chúng ta tiếp tục lấy các đỉnh từ trong ngăn xếp để tìm xem có còn bất kỳ đỉnh nào liền kề mà chưa được duyệt hay không. Trong ví dụ này là không có, và chúng ta vẫn tiếp tục cho tới khi ngăn xếp là trống.

# 2.Tìm kiếm theo chiều Ngang – Rộng BFS

Tìm kiếm theo chiều rộng (BFS) là một thuật toán tìm kiếm trong đồ thị trong đóviệc tìm kiếm chỉ bao gồm 2 thao tác: (a) cho trước một đỉnh của đồ thị; (b) thêmcác đỉnh kề với đỉnh vừa cho vào danh sách có thể hướng tới tiếp theo. Có thể sửdụng thuật toán tìm kiếm theo chiều rộng cho hai mục đích: tìm kiếm đường đi từmột đỉnh gốc cho trước tới một đỉnh đích, và tìm kiếm đường đi từ đỉnh gốc tới tấtcả các đỉnh khác. Trong đồ thị không có trọng số, thuật toán tìm kiếm theo chiềurộng luôn tìm ra đường đi ngắn nhất có thể. Thuật toán BFS bắt đầu từ đỉnh gốc vàlần lượt nhìn các đỉnh kề với đỉnh gốc. Sau đó, với mỗi đỉnh trong số đó, thuật toánlại lần lượt nhìn trước các đỉnh kề với nó mà chưa được quan sát trước đó và lặplại. Xem thêm thuật toán tìm kiếm theo chiều sâu, trong đó cũng sử dụng 2 thao táctrên nhưng có trình tự quan sát các đỉnh khác với thuật toán tìm kiếm theo chiềurộng. giải thuật tìm kiếm theo chiều rộng duyệt từ A tới B tới E tới F sau đó tới C, tới Gvà cuối cùng tới D. Giải thuật này tuân theo qui tắc:

*Quy tắc*

Qui tắc 1: Duyệt tiếp tới đỉnh liền kề mà chưa được duyệt. Đánh dấu đỉnh mà đã được duyệt. Hiển thị đỉnh đó và đẩy vào trong một hàng đợi (queue)..

Qui tắc 2: Nếu không tìm thấy đỉnh liền kề, thì xóa đỉnh đầu tiên trong hàng đợi.

Qui tắc 3: Lặp lại Qui tắc 1 và 2 cho tới khi hàng đợi là trống.

*Ưu nhược điểm*

Ưu điểm:

Xét duyệt tất cả các đỉnh để trả về kết quả

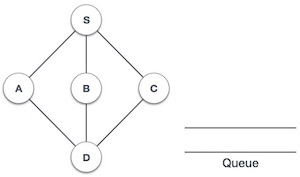
Nếu số đỉnh là hữu hạn, thuật toán chắc chắn tìm ra kết quả

Nhược điểm:

Mang tính chất vét cạn, không nên áp dụng nếu duyệt số đỉnh quá lớn.

Mang tính chất mù quáng, duyệt tất cả đỉnh, không chú ý đến thông tintrong các đỉnh để duyệt hiệu quả, dẫn đến duyệt qua các đỉnh khôngcần thiết

*Ví dụ:*



B1: Khởi tạo hàng đợi (queue)

B2: Chúng ta bắt đầu duyệt đỉnh S (đỉnh bắt đầu) và đánh dấu đỉnh này là đã duyệt.

B3: Sau đó chúng ta tìm đỉnh liền kề với Smà chưa được duyệt. Trong ví dụ này chúng ta có 3 đỉnh, và theo thứ tự chữ cái chúng ta chọn đỉnh A đánh dấu là đã duyệt và xếp A vào hàng đợi.

B4: Tiếp tục duyệt đỉnh liền kề với S là B. Đánh dấu là đã duyệt và xếp đỉnh này vào hàng đợi.

B5: Tiếp tục duyệt đỉnh liền kề với S là C. Đánh dấu là đã duyệt và xếp đỉnh này vào hàng đợi.

B6: Bây giờ đỉnh S không còn đỉnh nào liền kề mà chưa được duyệt. Bây giờ chúng ta rút A từ hàng đợi.

B7: Từ đỉnh A chúng ta có đỉnh liền kề là D và là đỉnh chưa được duyệt. Đánh dấu đỉnh D là đã duyệt và xếp vào hàng đợi.

# 3.Tìm kiếm Sâu dần

*Khái niệm*

Trong trí tuệ nhân tạo hay lý thuyết đồ thị, thuật toán tìm kiếm kiếm sâu dần là 1thuật toán duyệt hoặc tìm kiếm trên cây hoặc đồ thị.

Thuật toán được đưa ra để khắc phục điểm yêu của thuật toán tìm kiếm giới hạn độsâu DLS . Đó là khi mà tất cả các lời giải nằm ở độ sâu lớn hơn giới hạn độ sâu lthì giải thuậtDLS sẽ thất bại

*Thuật toán*

áp dụng giải thuật DLS đối với đường đi có độ dài <= 1

Nếu thất bại, tiếp tục áp dụng giải thuật dfs đối với đường đi có độ dài <= 2

..cứ như vậy đến khi tìm được lời giải hoặc khi toàn bộ cây đã được xétmà không tìm được lời giải

Luôn tìm ra nghiệm (nếu bài toán có nghiệm), miễn là chọn max đủ lớn(giống như tìm kiếm theo chiều rộng)

Có độ phức tạp thời gian là O(kd) (giống tìm kiếm rộng)

Có độ phức tạp không gian là O(k\*d) (giống tìm kiếm sâu)

Giải thuật tìm kiếm sâu dần thương áp dụng cho các bài toán có không giantrạng thái lớn và độ sâu của nghiệm không biết trước.

*Ưu nhược điểm*

Ưu điểm:

Nó tổ chức các lợi ích của thuật toán tìm kiếm BFS và DFS về mặt hiệu quảtìm kiếm và bộ nhớ nhanh.

Hạn chế chính của IDS là nó lặp lại tất cả các công việc của giai đoạn trước.

# 4.Tk Sâu có giới hạn DLS

Trong trí tuệ nhân tạo hay các lý thuyết đồ thị, thuật toán tìm kiếm có giới hạn độsâu (DLS) hay depth-limited search algorithm là một thuật toán phát triển các nútchưa xét các theo chiều sâu nhưng có giới hạn mức để tránh đi vào những conđường không mang lại kết quả tốt như trong thuật toán tìm kiếm sâu dần.

*Quy tắc*

*Ưu nhược điểm*

Ưu điêm:

Nó là bộ nhớ hiệu quả, sử dụng không gian tuyến tính O (bxL)

Nhược điểm:

Chưa hoàn thành nếu giải pháp nằm dưới giới hạn L(d <l), vì nókhông thể tìm thấy Solution.

Nó có thể không tìm thấy tối ưu nếu có nhiều hơn Solution.

Nó không hiệu quả về thời gian vì phải mất O (b ^ L)

Nó có thể gây ra các vòng lặp nếu tìm kiếm cây được sử dụng trên biểu đồ.

*Ví dụ*

# 5.Tìm kiếm leo đồi

*Quy tắc*

Tìm kiếm leo đồi là tìm kiếm theo độ sâu được hướng dẫn bởi hàm đánh giá. Songkhác với tìm kiếm theo độ sâu, khi phát triển một đỉnh u thì bước tiếp theo ta chọntrong số các đỉnh con của u, đỉnh có hứa hẹn nhiều nhất để phát triển, đỉnh nàyđược xác định bởi hàm đánh giá.

*Ưu nhược điểm*

Ưu điểm:

Phương pháp tìm kiếm leo đồi chú trọng tìm hướng đi dễ dẫn đến trạng tháiđích nhất. Cách đó được đưa ra nhằm làm giảm công sức tìm kiếm.

Thuậttoán tìm kiếm leo đồi thực chất là thuật toán tìm kiếm theo chiều sâu, songtại mỗi bước ta sẽ ưu tiên chọn một trạng thái có hứa hẹn nhanh tới đichnhất để phát triển trước.

Vấn đề quan trọng là biết khai thác kheo léo thôngtin phản hồi để xác định hướng đi tiếp và đẩy nhanh quá trình tìm kiếm.

Thông thường ta gán mỗi trạng thái của bài toán với một số đo (hàm đánh giá) nào đó nhằm đánh giá mức độ gần đích của nó.

Điều đó có nghĩa lànếu trạng thái hiện thời là u thì trạng thái v sẽ được phát triển tiếp theo nếuv kề với u và hàm đanh giá của v đạt giá trị max (hoặc min).

Nhược điểm:

Cực trị địa phương: nút đang xét tốt hơn các nút lân cận, nhưng đó không phải là phương án tốt nhất trong toàn thể, ví vậy có thể phải quay lui về núttrước để đi theo hướng khác. Giải pháp này đòi hỏi ghi nhớ lại nhiều đường đi.

Cao nguyên: Các giá trị của các phương án như nhau, không xác định đượcngay hướng nào là tốt hơn trong vùng lân cận.

*Ví dụ*

# 6.Tk Nhánh biên

# 7.Tk Tham ăn

*Khái niệm*

Giải thuật tham lam lựa chọn giải pháp nào được cho là tốt nhất ở thời điểm hiệntại và sau đó giải bài toán con nảy sinh từ việc thực hiện lựa chọn đó. Lựa chọn củagiải thuật tham lam có thể phụ thuộc vào lựa chọn trước đó. Việc quyết định sớmvà thay đổi hướng đi của giải thuật cùng với việc không bao giờ xét lại các quyếtđịnh cũ sẽ dẫn đến kết quả là giải thuật này không tối ưu để tìm giải pháp toàn cục.

*Ưu nhược điểm*

Ưu điểm:

Phân tích thời gian chạy cho các thuật toán tham lam nói chung sẽ dễ dànghơn nhiều so với các kỹ thuật khác (như Phân chia và chinh phục).

Đối vớikỹ thuật Phân chia và chinh phục, không rõ kỹ thuật này nhanh hay chậm. Điều này là do ở mỗi cấp độ đệ quy kích thước nhỏ hơn và số lượng các vấnđề phụ tăng lên.

Nhược điểm:

Điều khó khăn là đối với các thuật toán tham lam, bạn phải làm việc chămchỉ hơn nhiều để hiểu các vấn đề chính xác. Ngay cả với thuật toán chính xác, thật khó để chứng minh tại sao nó đúng.

Chứng minh rằng một thuậttoán tham lam là chính xác là một nghệ thuật hơn là một khoa học. Nó liênquan đến rất nhiều sự sáng tạo.

*Ví dụ*

# 8.Trình bài chuẩn tảc 8 quy tắc chuyển dạng logic vi từ, 4 quy tắc mệnh đề, cú pháp logic mênh đề

Nguyên lý, ưu nhược điểm, ví dụ, giải thích\

https://www.academia.edu/42920374/Khuongpro\_M%E1%BB%99t\_S%E1%BB%91\_Gi%E1%BA%A3i\_Thu%E1%BA%ADt\_M%C3%B4n\_Tr%C3%AD\_Tu%E1%BB%87\_Nh%C3%A2n\_T%E1%BA%A1o