***TỔNG HỢP | CÁC THUẬT TOÁN TÌM KIẾM | TRÍ TUỆ NHÂN TẠO***

***FROM VƯỢT CẠN | FIT.VIMARU.K58***

***Trong dạng 1 sẽ bao gồm 7 thuật toán tìm kiếm đường đi***

*1. Tìm kiếm theo chiều rộng*

*2. Tìm kiếm theo chiều sâu*

*3. Tìm kiếm tốt nhất đầu tiên = Tìm kiếm theo chiều rộng + Trọng số*

*4. Tìm kiếm leo đồi = Tìm kiếm theo chiếu sâu + Trọng số*

*5. Thuật toán A\* = Tìm kiếm tốt nhất đầu tiên + Hàm đánh giá*

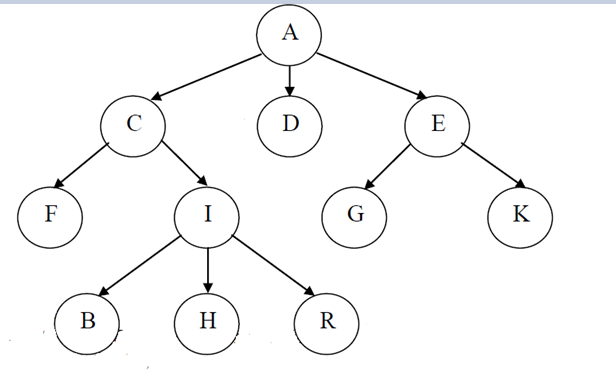
*6. Thuật toán nhánh cận = Tìm kiếm leo đồi + Hàm đánh giá*

*7. Chiến thuật minimax*

**Chi tiết các thuật toán tìm kiếm :**

**1. Tìm kiếm theo chiều rộng / chiều sâu :**

VD :



**1.1 Chiều rộng (Vào trước ra trước)**

|  |  |  |
| --- | --- | --- |
| U | V | L |
| A | C,D,E | C,D,E |
| C | F,I | D,E,F,I |
| D | Rỗng | E,F,I |
| E | G,K | F,I,G,K |
| F | Rỗng | I,G,K |
| I | B,H,R | G,K,B,H,R |
| G | Rỗng | K,B,H,R |
| K | Rỗng | B,H,R |
| B | Rỗng | H,R |
| H | Rỗng | R |
| R | Rỗng | Rỗng |
| END |  |  |

**1.2. Chiều sâu : (Vào sau ra trước)**

|  |  |  |
| --- | --- | --- |
| U | V | L |
| A | C,D,E | C,D,E |
| C | F,I | F,I,D,E |
| F | Rỗng | I,D,E |
| I | B,H,R | B,H,R,D,E |
| B | Rỗng | H,R,D,E |
| H | Rỗng | R,D,E |
| R | Rỗng | D,E |
| D | Rỗng | E |
| E | G,K | G,K |
| G | Rỗng | K |
| K | Rỗng | Rỗng |
| END |  |  |

**2. Tìm kiếm tốt nhất đầu tiên / leo đồi**

20

A

18

15

16

E

D

C

12

6

K

G

I

F

10

8

R

H

M

5

6

4

0

B

**2.1. Tìm kiếm tốt nhất đầu tiên :**

|  |  |  |
| --- | --- | --- |
| U | V | L |
| A(20) | C(16),D(15),E(18) | D(15),C(16),E(18) |
| D(15) | RỖNG | C(16);E(18) |
| C(16) | F(8);I(12) | F(8);I(12);E(18) |
| F(8) | M(4) | M(4);I(12);E(18) |
| M(4) | B(0) | B(0);I(12);E(18) |
| B(0) | ĐÍCH,END |  |

**2.2. Tìm kiếm leo đồi**

|  |  |  |  |
| --- | --- | --- | --- |
| U | V | L1 | L |
| A(20) | C(16),D(15),E(18) | D(15),C(16),E(18) | D(15),C(16),E(18) |
| D(15) | RỖNG | RỖNG | C(16);E(18) |
| C(16) | F(8);I(12) | F(8);I(12) | F(8);I(12);E(18) |
| F(8) | M(4) | M(4) | M(4);I(12);E(18) |
| M(4) | B(0) | B(0) | B(0);I(12);E(18) |
| B(0) | ĐÍCH,END |  |  |

(Lưu ý : Chỉ sắp xép L1 theo thứ tự tăng dần, rồi đẩy vào đầu danh sách L, **KHÔNG SẮP XẾP LẠI L)**

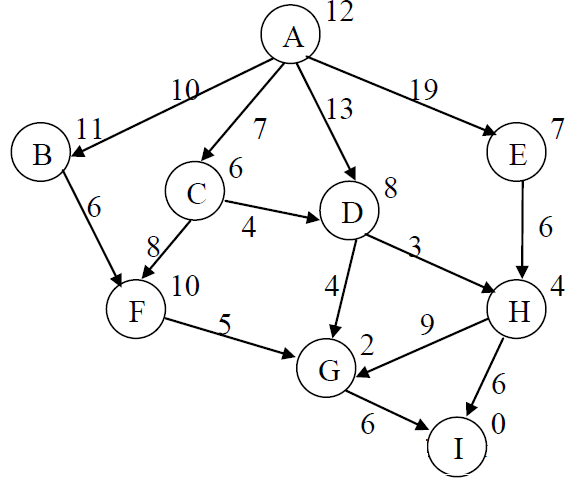
**3. Tìm kiếm A\* / Nhánh cận**

Ở hai phương pháp tìm kiếm này, xuất hiện một đối tượng mới – hàm đánh giá

Hàm đánh giá : f(u) = g(u) + h(u)

Trong đó : g(u) – tổng trọng số cạnh nối từ đỉnh xuất phát tới đỉnh đang xét (u)

h(u) – trọng số của đỉnh đang xét

VD : 

**3.1. Thuật toán A\* :**

(Trình bày chi tiết cho anh em hiểu rỗ cách tính hàm đánh giá)

Khởi tạo L để lưu các đỉnh được chọn

L = {A}

* Phát triển đỉnh A sinh ra các đỉnh con là B,C,D,E có :

g(B) = 10, f(B) = g(B) + h(B) = 10 + 11 = 21

g(C) = 7, f(C) = g(C) + h(C) = 7 + 6 = 13

g(D) = 13, f(D) = g(D) + h(D) = 13 + 8 = 21

g(E) = 19, f(E) = g(E) + h(E) = 19 + 7 = 26

L = {A,C}

* Phát triển đỉnh C vì f(C) nhỏ nhất sinh ra các đỉnh con là D,F có :

g(D) = g(C) + k(C,D) = 7 + 4 = 11, f(D) = g(D) + h(D) = 11 + 8 = 19

g(F) = g(C) + k(C,F) = 7 + 8 = 15, f(F) = g(F) + h(F) = 15 + 10 = 25

L = {A,C,D}

* Phát triển đỉnh D vì f(D) nhỏ nhất sinh ra các đỉnh con là G,H có :

g(G) = g(D) + k(D,G) = 11 + 4 = 15, f(G) = g(G) + h(G) = 15 + 2 = 17

g(H) = g(D) + k(D,H) = 11 + 3 = 14, f(H) = g(H) + h(H) = 14 + 4 = 18

L = {A,C,D,G}

* Phát triển đỉnh G vì f(G) nhỏ nhất sinh ra các đỉnh con là I có :

g(I) = g(G) + k(G,I) = 15 + 6 = 21, f(I) = g(I) + h(I) = 21 + 0 = 21

L = {A,C,D,G,I}

|  |  |  |
| --- | --- | --- |
| U | V | L |
| A(12) | B(21);C(13);D(21);E(26) | C(13);B(21);D(21);E(26) |
| C(13) | D(19);F(25) | D(19);B(21);D(21);F(25);E(26) |
| D(19) | G(17);H(18) | G(17);H(18);B(21);D(21);F(25);E(26) |
| G(17) | I(21) | H(18);B(21);D(21);I(21);F(25);E(26) |
| H(18) | G(25),I(20) | I(20); B(21);D(21);I(21);F(25);G(25);E(26) |
| I(20 | DICH,END |  |

**Ở thuật toán A\*, chỉ cần tìm thấy đích, lập tức dừng lại**

Đường đi cuối cùng sẽ là : A-C-D-G-I với chi phí là f(I) = 21

**3.2. Thuật toán nhánh cận ( Tương tự với leo đồi )**

Ở thuật toán này, vì có tính chất “leo đồi”, nên sẽ phải tìm đến đích với đường đi mà có tổng trọng số f là nhỏ nhất, và để tìm được đích có f nhỏ nhất, thì buộc phải duyệt cả cây

Bài này nếu duyệt hết cả sơ đồ sẽ rất dài, vì vậy, anh em nên bỏ thời gian vẽ sơ đồ cây ra trước để xác định xem trọng số các đỉnh ở đích là bao nhiêu, từ đó dừng bài làm ở đỉnh có trọng số bé nhất, tránh mất nhiều thời gian phải duyệt cả cây

|  |  |  |  |
| --- | --- | --- | --- |
| U | V | L1 | L |
| A(12) | B(21);C(13);D(21);E(26) | C(13);B(21);D(21);E(26) | C(13);B(21);D(21);E(26) |
| C(13) | D(19);F(25) | D(19);F(25) | D(19);F(25); B(21);D(21);E(26) |
| D(19) | G(17);H(18) | G(17);H(18) | G(17);H(18);F(25); B(21);D(21);E(26) |
| G(17) | I(21) | I(21) | I(21);H(18);F(25); B(21);D(21);E(26) |
| I(21) | RONG | RONG | H(18);F(25); B(21);D(21);E(26) |

Tôi chỉ duyệt 5 lần để anh em hiểu cách duyệt của thuật toán này, kết quả cuối cùng, đích sẽ là I(17) với đường đi là A – D – G – I, qua đâu đó gần 30 phép duyệt để duyệt toàn bộ các đỉnh, các đường đi

**4. Chiến lược Minimax :**

Min

Max

Min

C

A

D

E

F

I

G

B

H

R

K

Max

-3

-2

4

-3

5

5

3

Trả lời : Thứ tự tính toán như sau :

I ở mức Max => điểm số của I = max(B,H.R) = max(4,-3,5) = 5

C ở mức Min => điểm số của C = min(F,I) = min(-2,5) = -2

E ở mức Min => điểm số của E = min(G,K) = min(5,3) = 3

A ở mức Max => điểm số của A = max(C,D,E) = max(-2,-3,3) = 3

Vậy nước đi của quân A theo thuật toán tìm kiếm có đối thủ MinMax là A => E