

# A\* Search

---



# A\* Search

---



UCS



Greedy



A\*

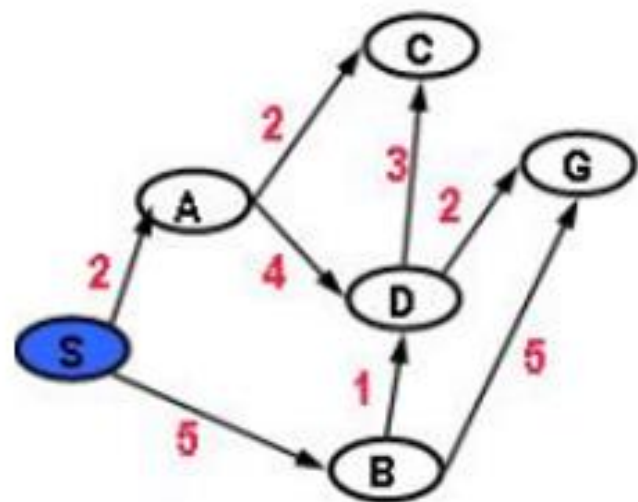
# A\* Search

- Greedy Search không đảm bảo tìm ra đường đi ngắn nhất là do thuật toán chỉ quan tâm tới chi phí ước lượng từ một nút tới đích mà không quan tâm tới chi phí đã đi từ nút xuất phát tới nút đó.
- Ý tưởng A\* Search: Sử dụng hàm đánh giá  $f(N) = g(N) + h(N)$ 
  - $g(N)$  = chi phí từ node gốc (Start state) cho đến node hiện tại N
  - $h(N)$  = chi phí ước lượng từ nút hiện tại n tới đích
  - $f(N)$  = chi phí tổng thể ước lượng của đường đi qua nút hiện tại N đến đích



# A\* Search

Ví dụ:



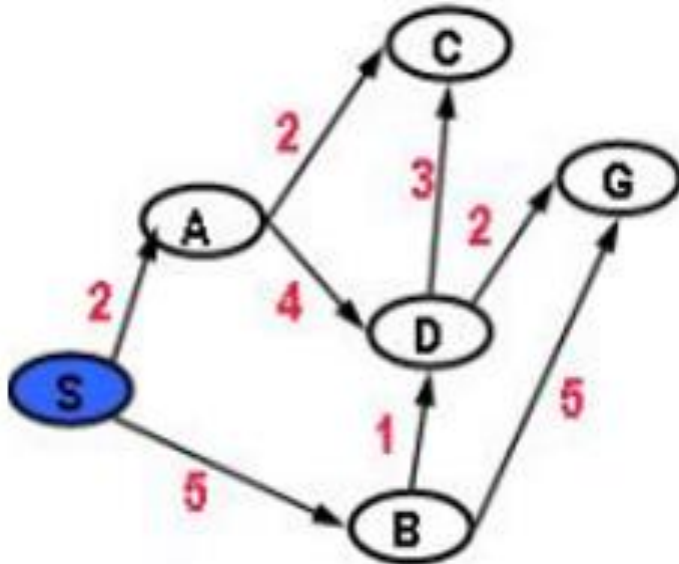
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	

# A\* Search

Ví dụ:



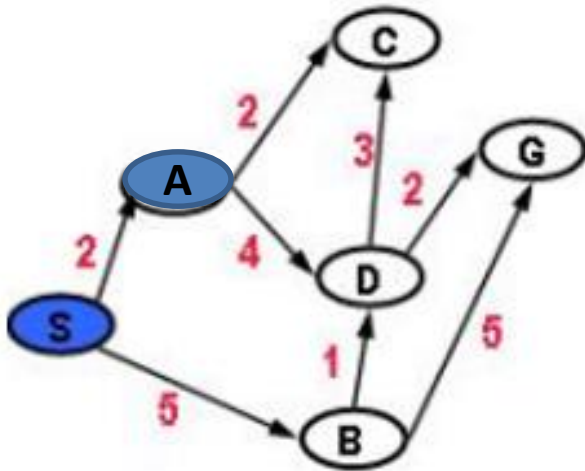
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	<u>(4,AS)</u> <u>(8,BS)</u>

# A\* Search

Ví dụ:



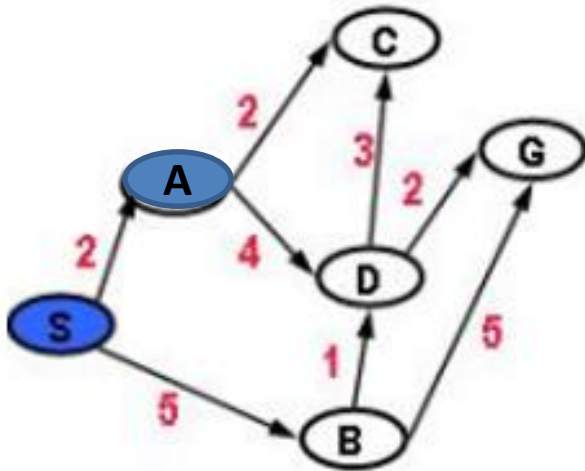
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	

# A\* Search

Ví dụ:



Heuristic Values

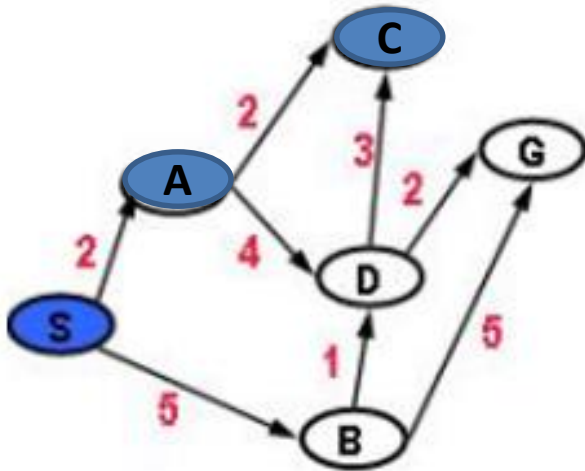
A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	<u>(5, CAS)</u> <u>(7,DAS)</u> (8,BS)



# A\* Search

Ví dụ:



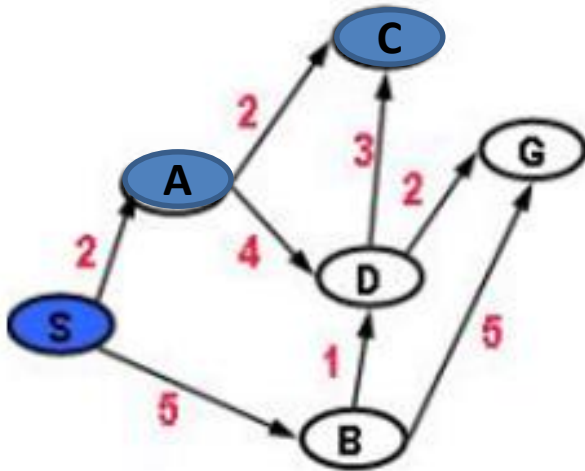
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	

# A\* Search

Ví dụ:



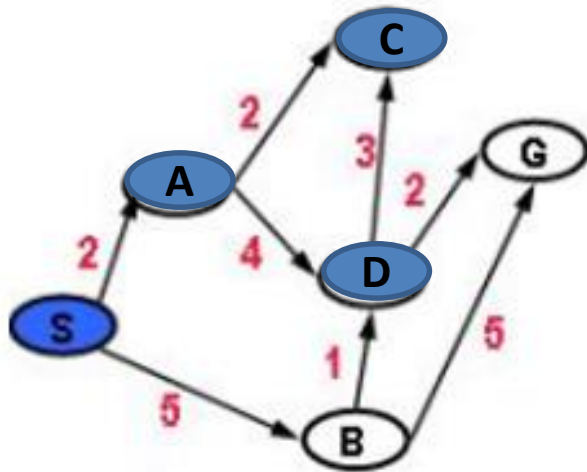
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)

# A\* Search

Ví dụ:



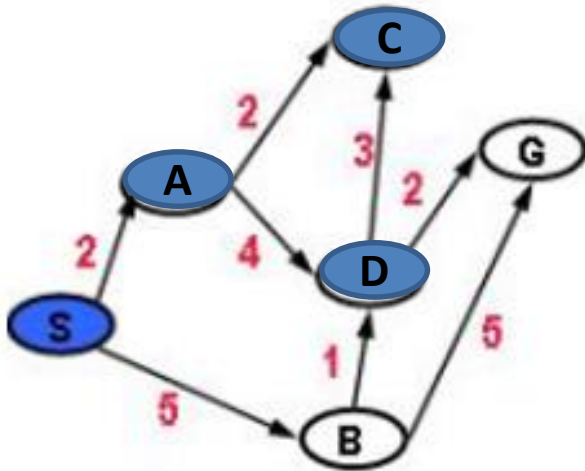
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	

# A\* Search

Ví dụ:



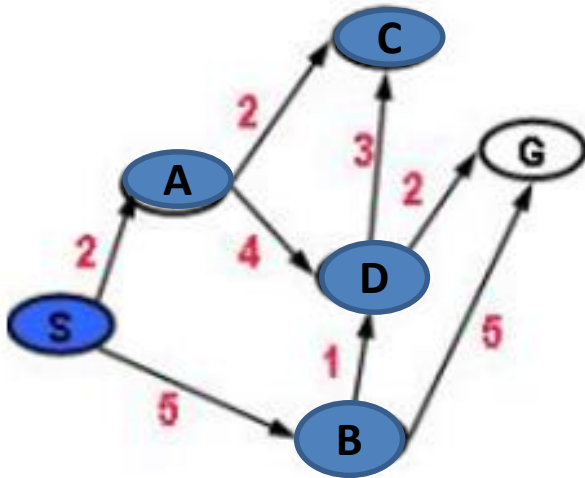
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	<u>(8,GDAS) (10,CDAS)</u> (8,BS)

# A\* Search

Ví dụ:



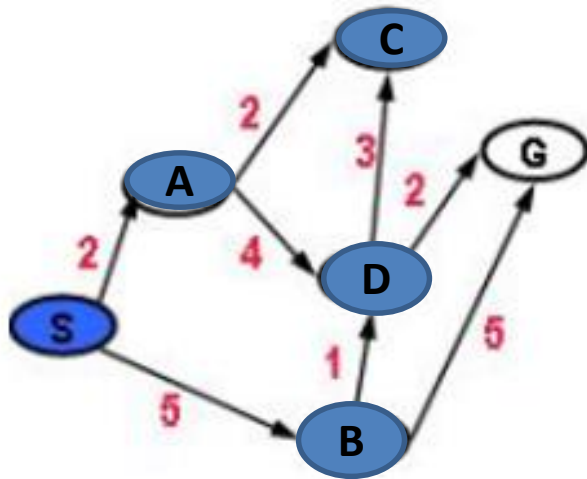
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	(8,GDAS) (10,CDAS)
(8,BS)	

# A\* Search

Ví dụ:



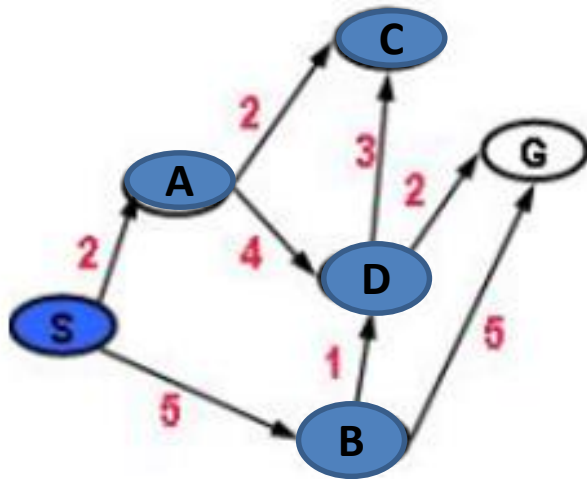
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	(8,GDAS) (10,CDAS)
(8,BS)	<u>(10,GBS), (7, DBS)</u> (8,GDAS) (10,CDAS)

# A\* Search

Ví dụ:



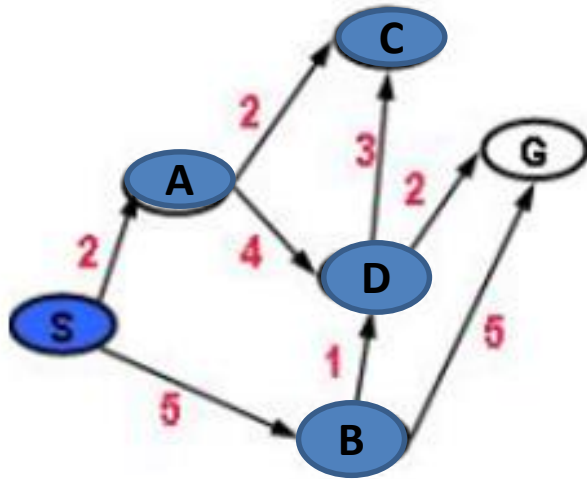
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	(8,GDAS) (10,CDAS)
(8,BS)	(10,GBS) (7, DBS) (8,GDAS) (10,CDAS)
(7, DBS)	

# A\* Search

Ví dụ:



Heuristic Values

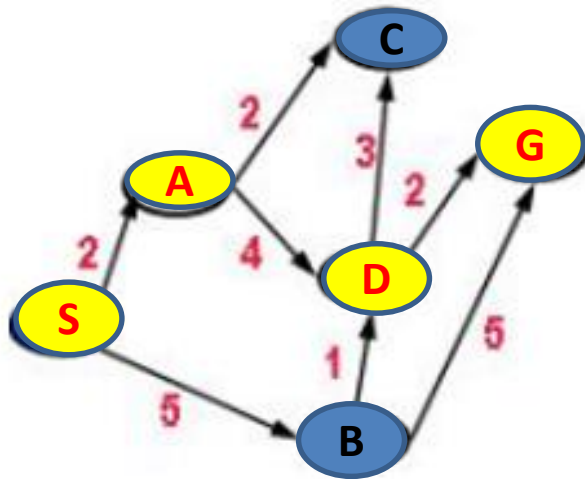
A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	(8,GDAS) (10,CDAS)
(8,BS)	(10,GBS) (7, DBS) (8,GDAS) (10,CDAS)
(7, DBS)	<u>(8,GDBS) (10, CDBS)</u> (10,GBS) (8,GDAS) (10,CDAS)



# A\* Search

Ví dụ:



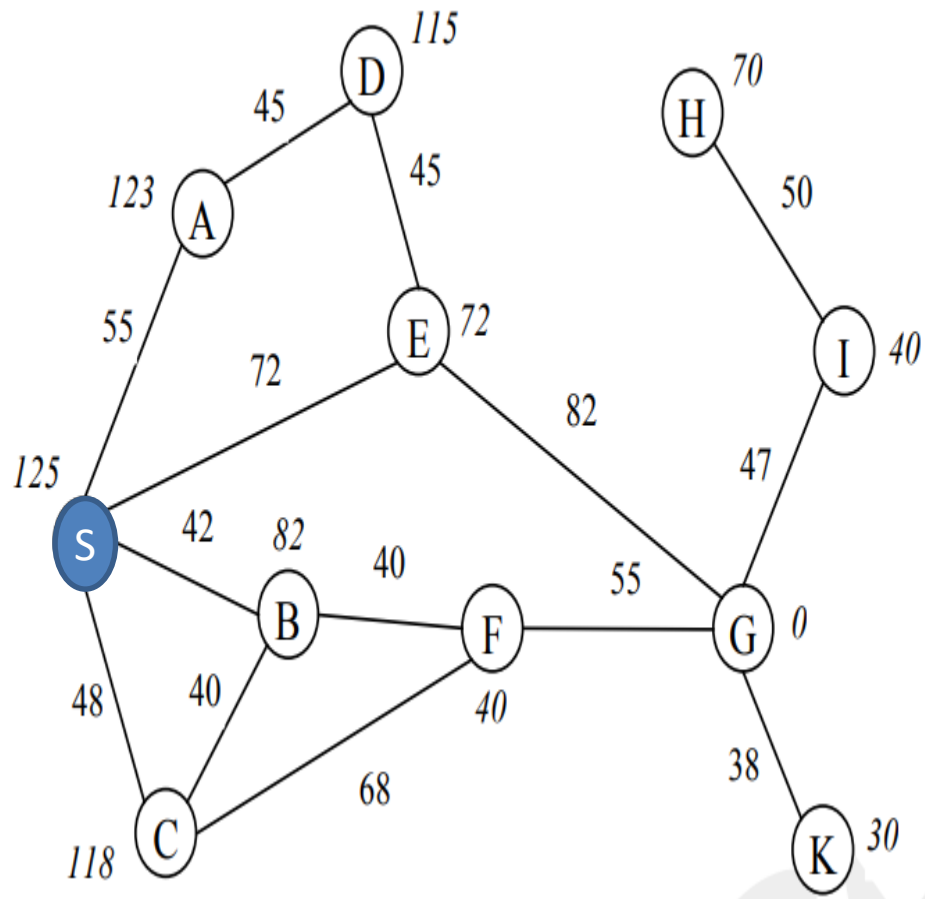
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node được xét	Tập biên O
	(0, S)
(0, S)	(4,AS) (8,BS)
(4,AS)	(5, CAS) (7,DAS) (8,BS)
(5, CAS)	(7,DAS) (8, BS)
(7,DAS)	(8,GDAS) (10,CDAS)
(8,BS)	(10,GBS) (7, DBS) (8,GDAS) (10,CDAS)
(7, DBS)	(8,GDBS) (10, CDBS) (10,GBS) (8,GDAS) (10,CDAS)
(8,GDAS)	ĐÍCH

# A\* Search

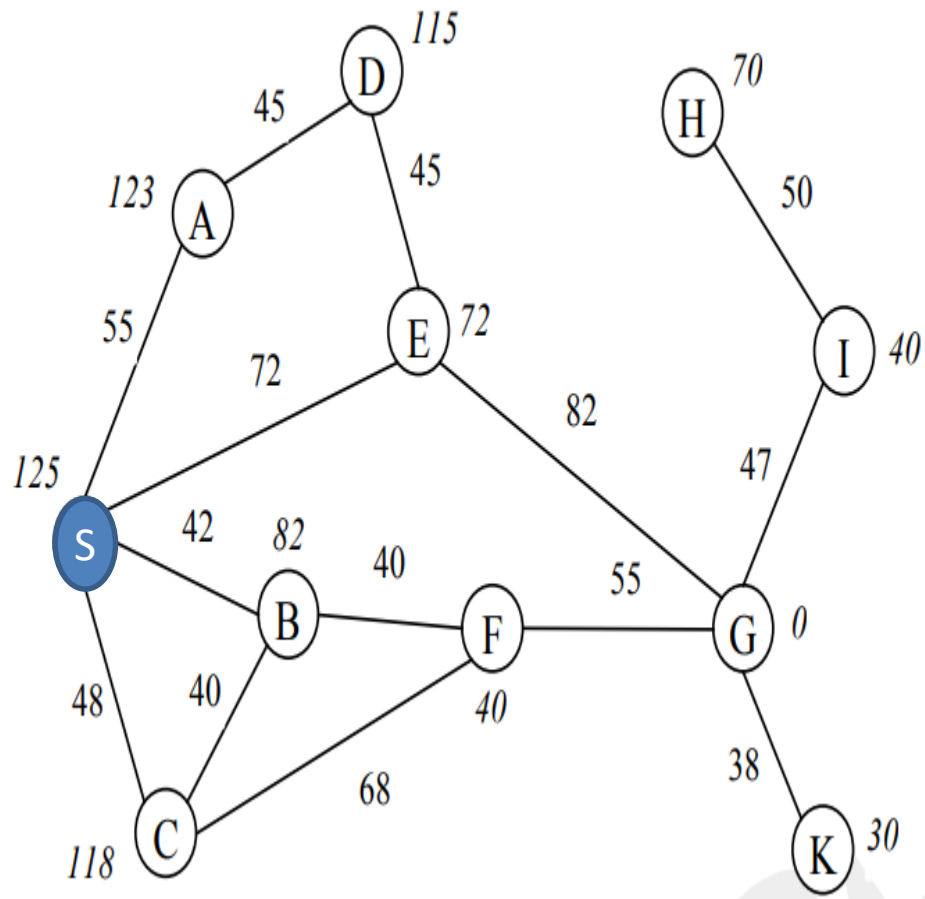
Ví dụ:



Node được xét	Tập biên O
	<u>(125, S)</u>

# A\* Search

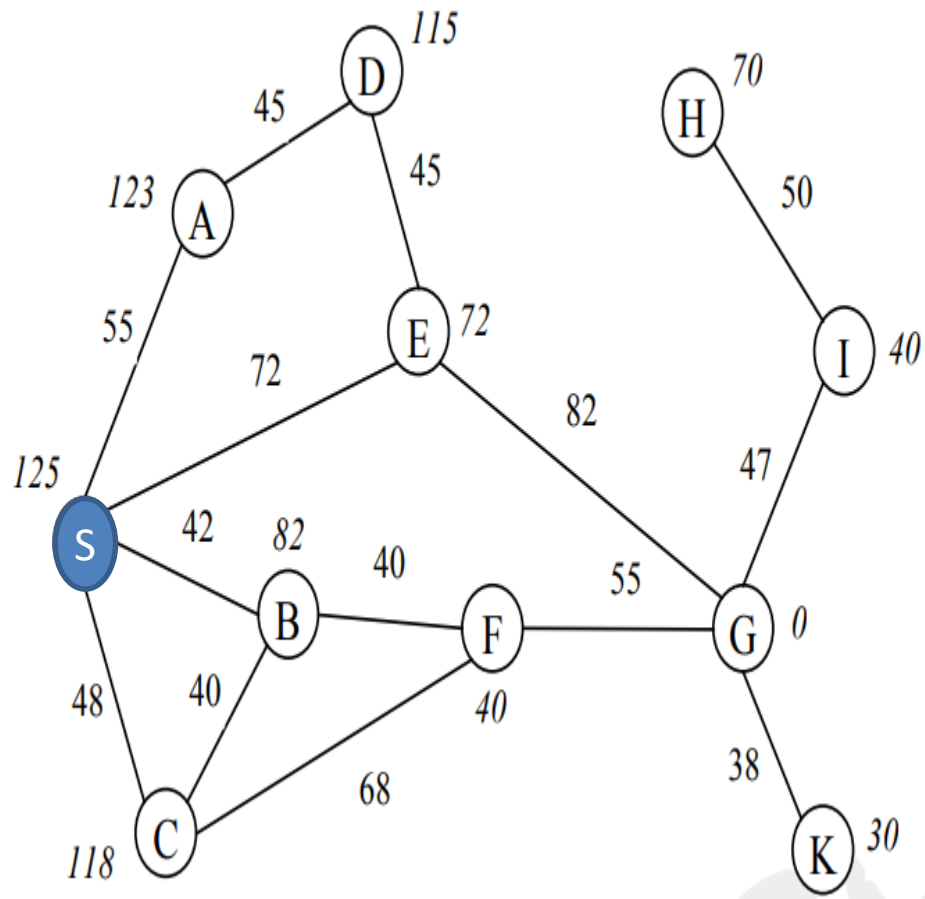
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	

# A\* Search

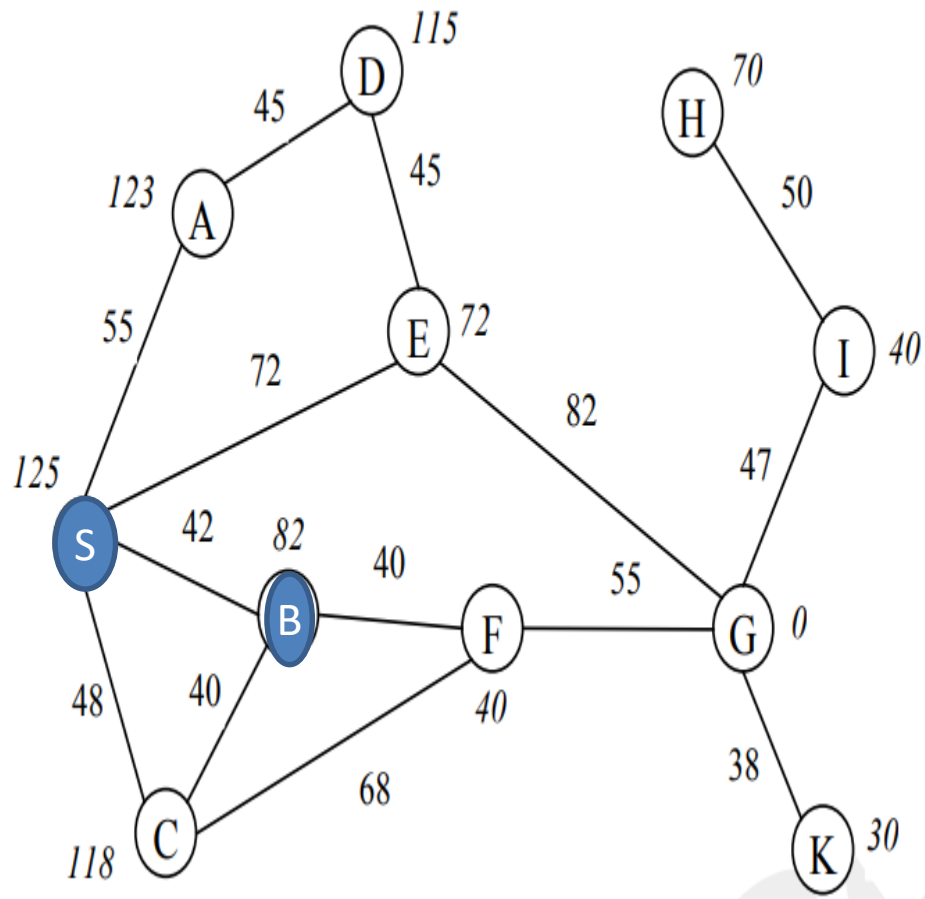
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	<u>(178,AS) (124,BS) (166,CS) (144,ES)</u>

# A\* Search

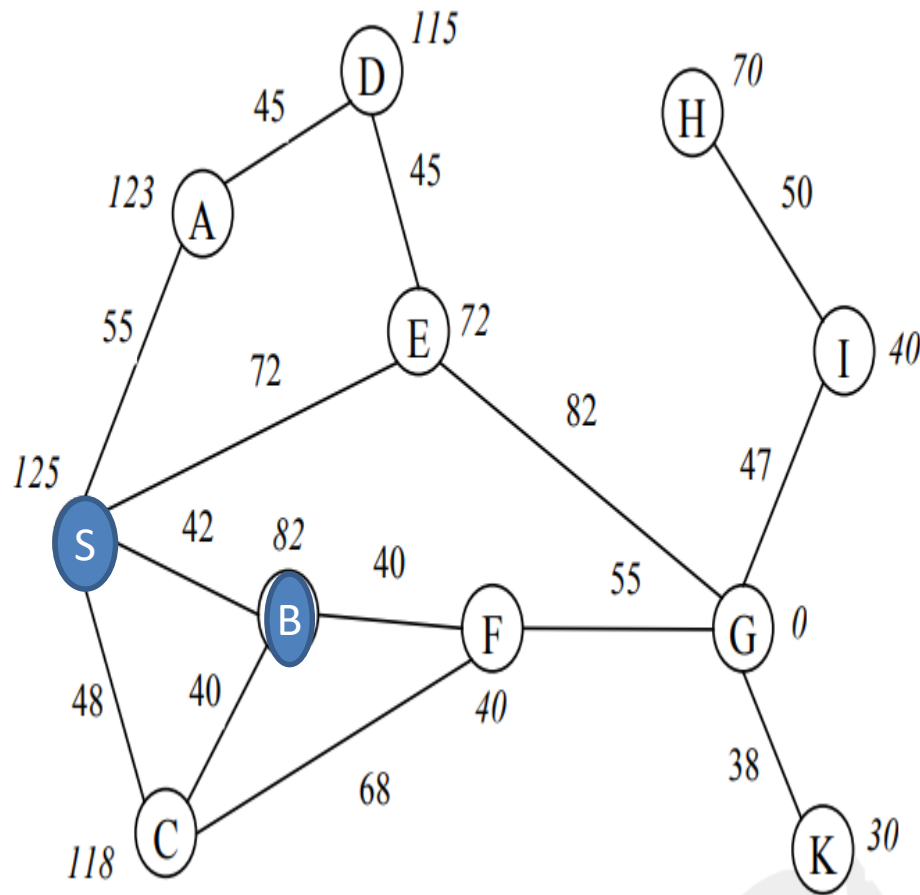
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	

# A\* Search

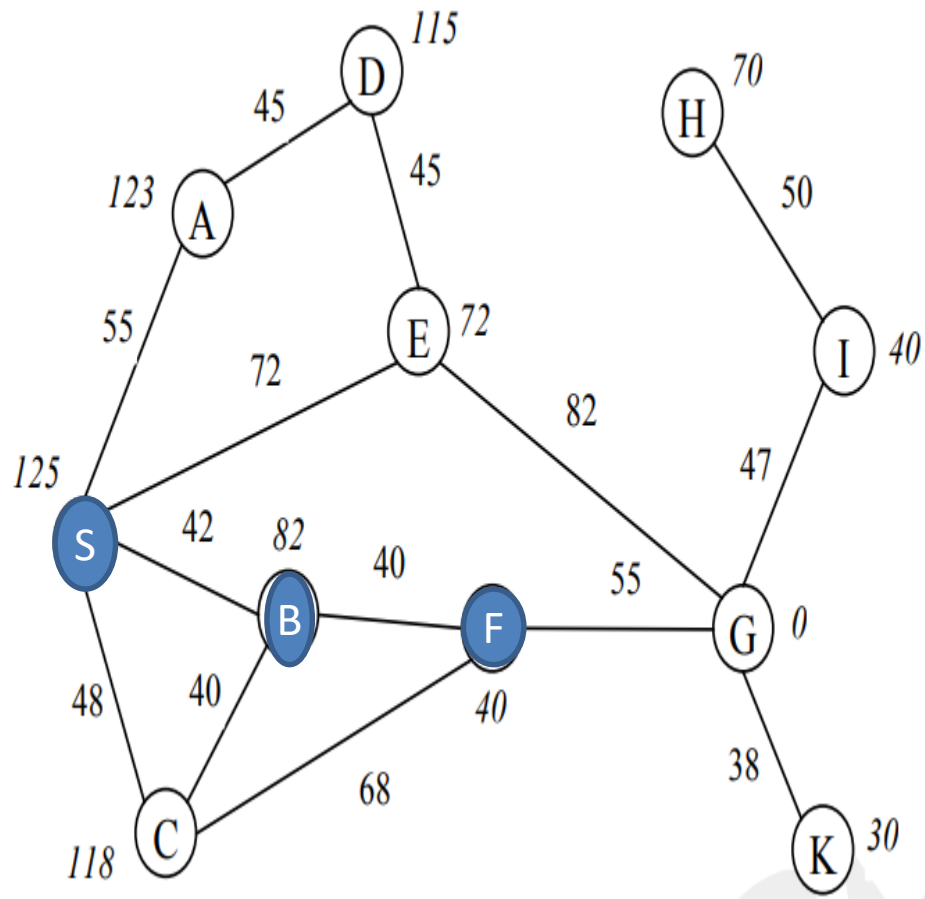
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>

# A\* Search

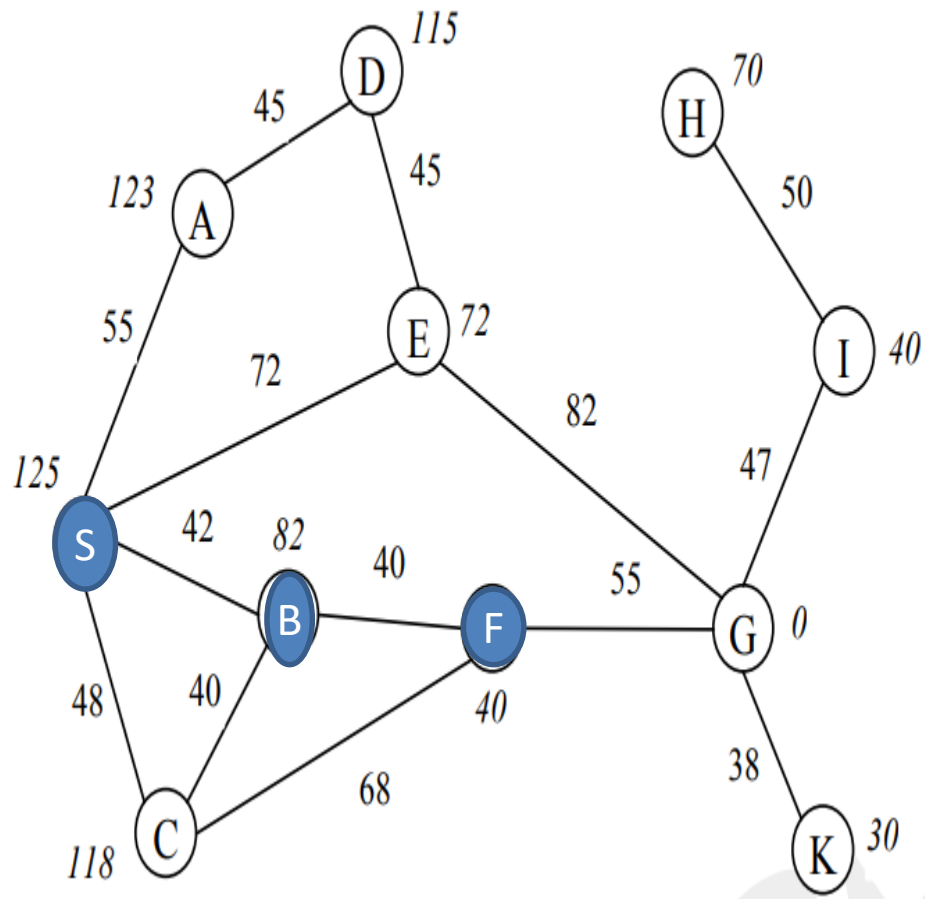
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) (200,CBS) (122,FBS)
(122,FBS)	

# A\* Search

Ví dụ:

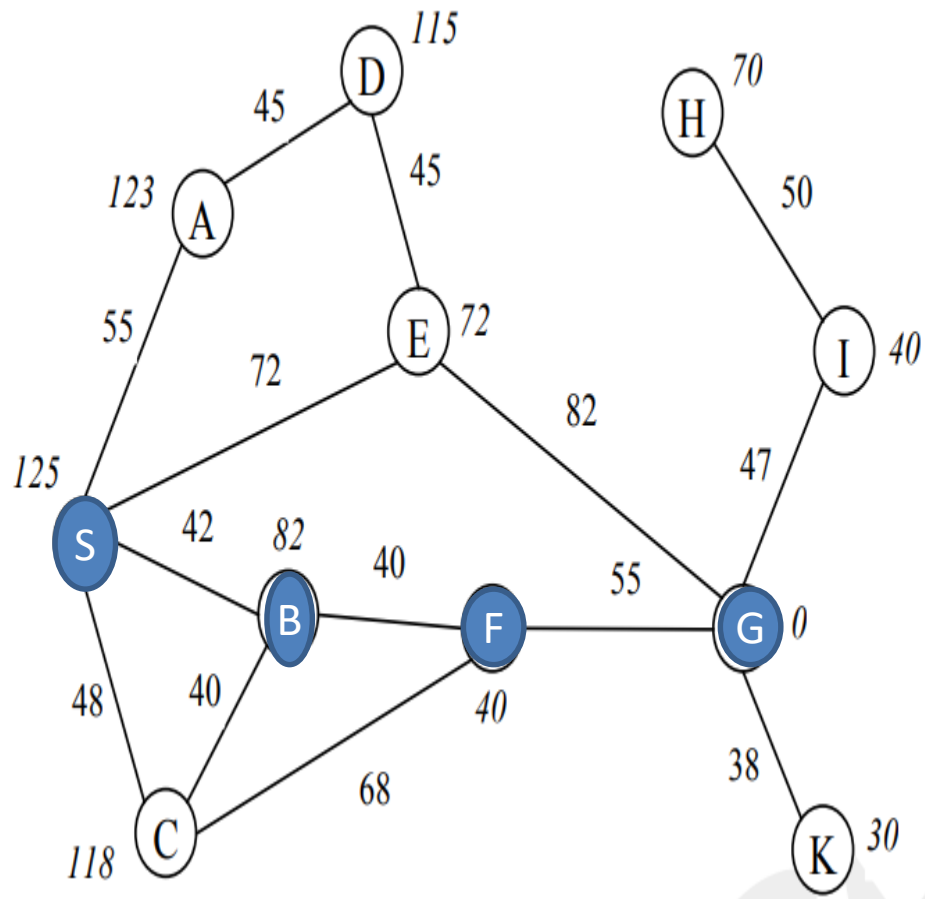


Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) (200,CBS) (122,FBS)
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) <u>(137,G,FBS) (268,C,FBS)</u>



# A\* Search

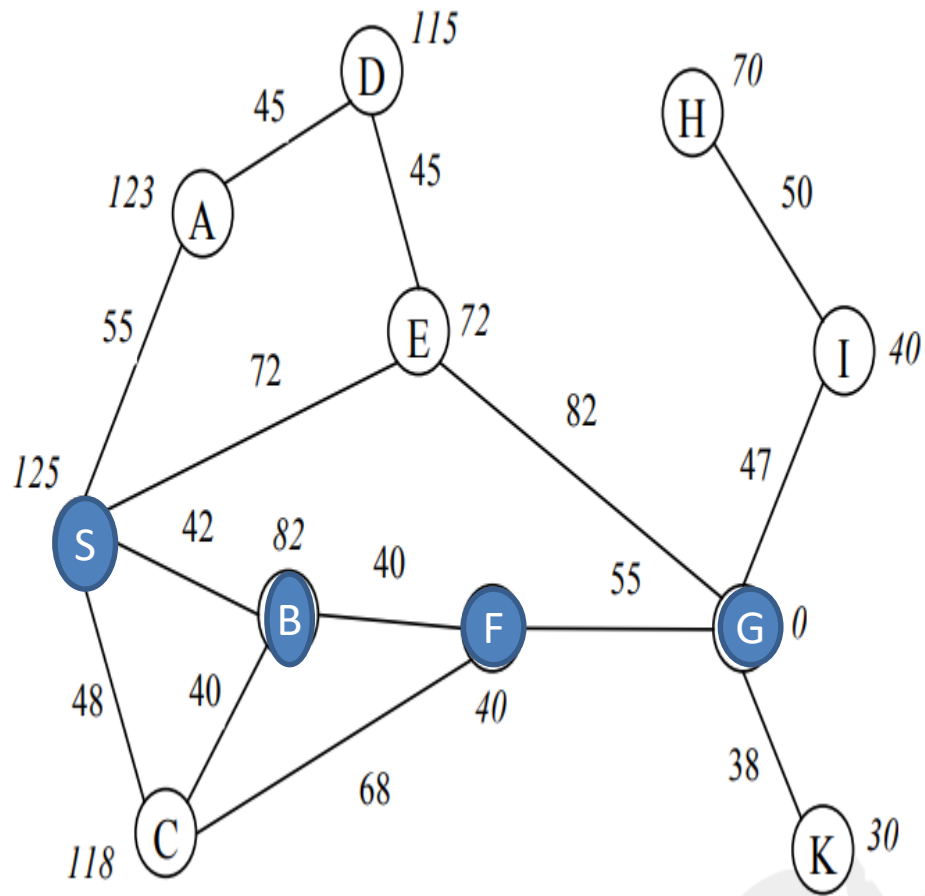
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) (200,CBS) (122,FBS)
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) (137,GFBS) (268,CFBS)
(137,GFBS)	

# A\* Search

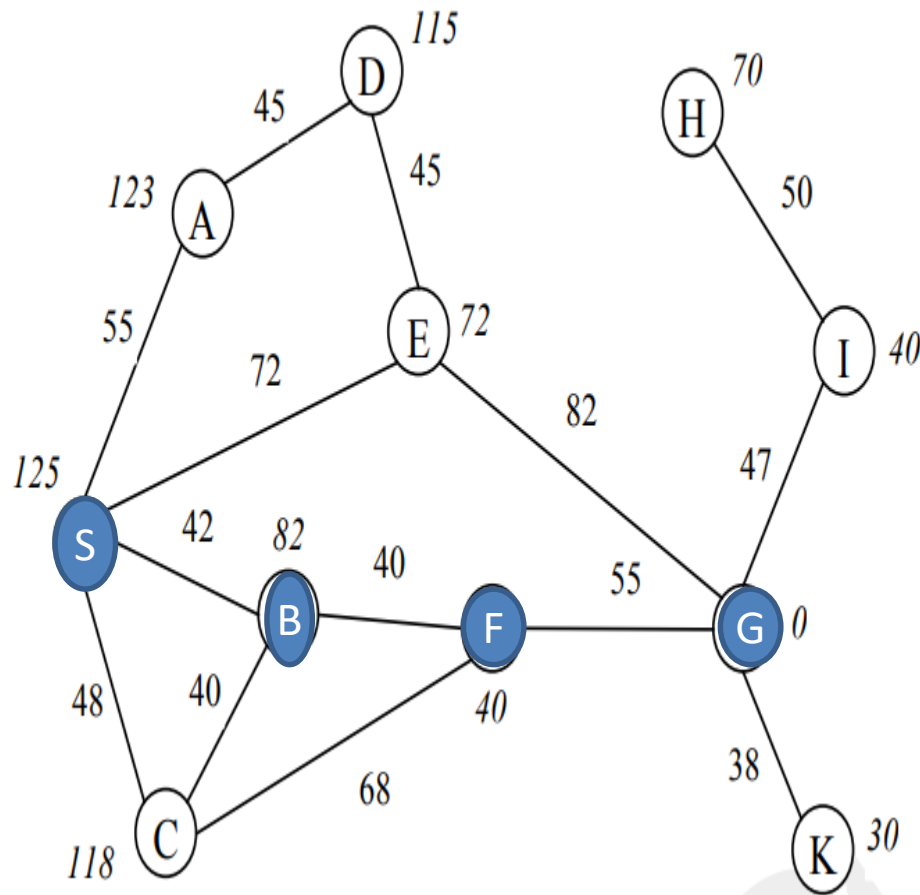
Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) (200,CBS) (122,FBS)
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) (137,GFBS) (268,CFBS)
(137,GFBS)	(178,AS) (166,CS) (144,ES) (200,CBS) (268,CFBS)

# A\* Search

Ví dụ:



Node được xét	Tập biên O
	(125, S)
(125, S)	(178,AS) (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) (200,CBS) (122,FBS)
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) (137,GFBS) (268,CFBS)
(137,GFBS)	ĐÍCH

Chúng ta đã tìm ra con đường (S->B->F->G) với chi phí bằng 137

# A\* Search

$A^*(O, S, G, P, h)$

- Đầu vào: bài toán tìm kiếm, hàm heuristic  $h$
- Đầu ra: đường đi ngắn nhất từ nút xuất phát đến nút đích
- Khởi tạo: tập các nút biên (nút mở)  $O \leftarrow S$

-----  
While( $O$  không rỗng) do

1. Lấy nút  $N$  có  $f(N)$  nhỏ nhất ra khỏi  $O$
2. Nếu  $n$  thuộc  $G$ , return(đường đi tới  $N$ )
3. Với mọi  $M \in P(n)$ 
  - i.  $f(M) = g(M) + h(M)$
  - ii. Thêm  $M$  vào  $O$  cùng giá trị  $f(M)$

Return: không tìm được đường đi

# A\* Search

- Đặc điểm:
  - Thuật toán cho kết quả tối ưu nếu hàm heuristic  $h$  là hàm chấp nhận được.
  - Thuật toán đầy đủ, trừ trường hợp có vô số các nút với hàm  $f$  có giá trị rất nhỏ nằm giữa node xuất phát và node đích.
  - Độ phức tạp: trong trường hợp xấu nhất, khi hàm heuristic không có nhiều thông tin, độ phức tạp tính toán và yêu cầu bộ nhớ của A\* đều là  $O(b^m)$ .
  - Trong tất cả các thuật toán tìm kiếm tối ưu sử dụng cùng hàm heuristics thì thuật toán A\* có độ phức tạp tính toán nhỏ nhất, tức là yêu cầu sinh ra ít nút nhất trước khi tìm ra lời giải.

Như thế nào là hàm heuristic chấp nhận được???

# A\* Search

- Hàm heuristic  $h(n)$  được gọi là chấp nhận được khi:  $h(n) \leq h^*(n)$ 
  - $h^*(n)$  là giá thành đường đi thực tế từ  $n$  đến node đích
  - hàm  $h(n)=0$  với mọi  $n$ , là hàm chấp nhận được
- Các hàm heuristic:
  - Được xây dựng tùy thuộc vào bài toán cụ thể.
  - Có thể có rất nhiều hàm heuristic khác nhau cho cùng một loại bài toán.
  - Chất lượng hàm heuristic ảnh hưởng rất nhiều đến quá trình tìm kiếm.
  - Trong tất cả các thuật toán tìm kiếm tối ưu sử dụng cùng hàm heuristics thì thuật toán A\* có độ phức tạp tính toán nhỏ nhất, tức là yêu cầu sinh ra ít nút nhất trước khi tìm ra lời giải.

# A\* Search

- Ví dụ:

3	1	6
5		4
2	7	8

Trạng thái hiện thời

	1	2
3	4	5
6	7	8

Trạng thái đích

- Trong bài toán 8 ô, có thể xây dựng một số hàm heuristic. Dưới đây, ta sẽ xem xét hai trong số các hàm như vậy.

# A\* Search

- Ta có thể sử dụng hai hàm heuristic sau:
  - $h1(n)$ : số ô đặt sai chỗ (ví dụ:  $h1(u)=5$ )
  - $h2(n)$ : tổng khoảng cách Manhattan giữa vị trí hiện thời của mỗi ô tới vị trí đúng của ô đó. (ví dụ:  $h2(u) = 1 + 4 + 1 + 2 + 1$ )
    - Khoảng cách Manhattan được tính bằng số ít nhất các dịch chuyển theo hàng hoặc cột để đưa một ô tới vị trí của nó trong trạng thái đích.
  - $h3(n)$ ????

3	1	6
5		4
2	7	8

Trạng thái hiện thời

	1	2
3	4	5
6	7	8

Trạng thái đích



# A\* Search

- Các ước lượng chấp nhận được

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(S) = \text{?????}$
- $h_2(S) = \text{?????}$
- $h_4(S) = \text{?????}$

(Số ô sai, khoảng cách và số ô cản trên đường dịch chuyển

# A\* Search

- Các ước lượng chấp nhận được

- $h_1(u)$  = Số lượng ô sai vị trí.
- $h_2(u)$  = Tổng khoảng cách theo Mahattan Metric

(i.e., Số lượng ô từ ô hiện tại đến vị trí mong muốn)

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(S) = 8$
- $h_2(S) = 3+1+2+2+2+3+3+2 = 18$
- $h_3(S) = 5+2+3+3+....$

(Số ô sai, khoảng cách và số ô cản trên đường dịch chuyển

# A\* Search

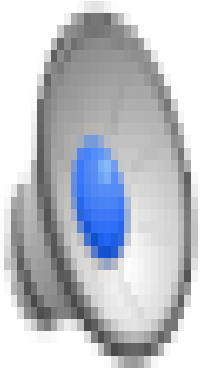
- Hàm heuristic trội:
  - Hàm tốt hơn, hàm nhanh dẫn tới kết quả hơn
  - Giả sử có hai hàm heuristic chấp nhận được  $h_1(n)$  và  $h_2(n)$ .
    - Nếu  $h_1(n) \leq h_2(n)$  với mọi  $n$  thì ta nói rằng  $h_2(n)$  trội hơn so với  $h_1(n)$
  - Trong trường hợp trong hai hàm  $h_1(n)$  và  $h_2(n)$  không có hàm trội hơn???

# A\* Search

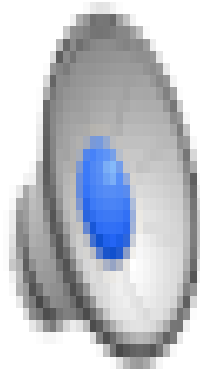
- Hàm heuristic trội:
  - Hàm tốt hơn, hàm nhanh dẫn tới kết quả hơn
  - Giả sử có hai hàm heuristic chấp nhận được  $h_1(n)$  và  $h_2(n)$ .
    - Nếu  $h_1(n) \leq h_2(n)$  với mọi  $n$  thì ta nói rằng  $h_2(n)$  trội hơn so với  $h_1(n)$
  - Trong trường hợp trong hai hàm  $h_1(n)$  và  $h_2(n)$  không có hàm trội hơn???
  - Có thể tạo ra hàm  $h'(n) = \max(h_1(n), h_2(n))$  với mọi  $n$ .

# So sánh giữa các thuật toán tìm kiếm

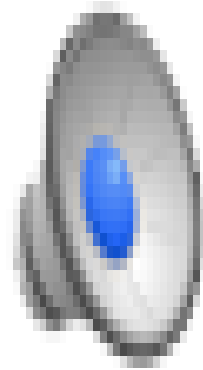
- Video of Demo Contours (Empty)



UCS



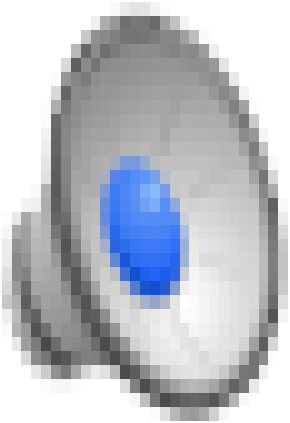
Greedy



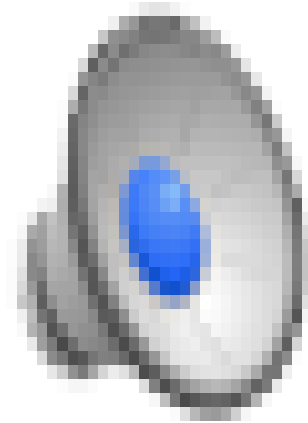
A\*

# So sánh giữa các thuật toán tìm kiếm

- Video of Demo Contours (Pacman Small Maze)



Greedy



A\*

# So sánh giữa các thuật toán tìm kiếm



Greedy



Uniform Cost



A\*

# A\* search

## Các ứng dụng của A\*

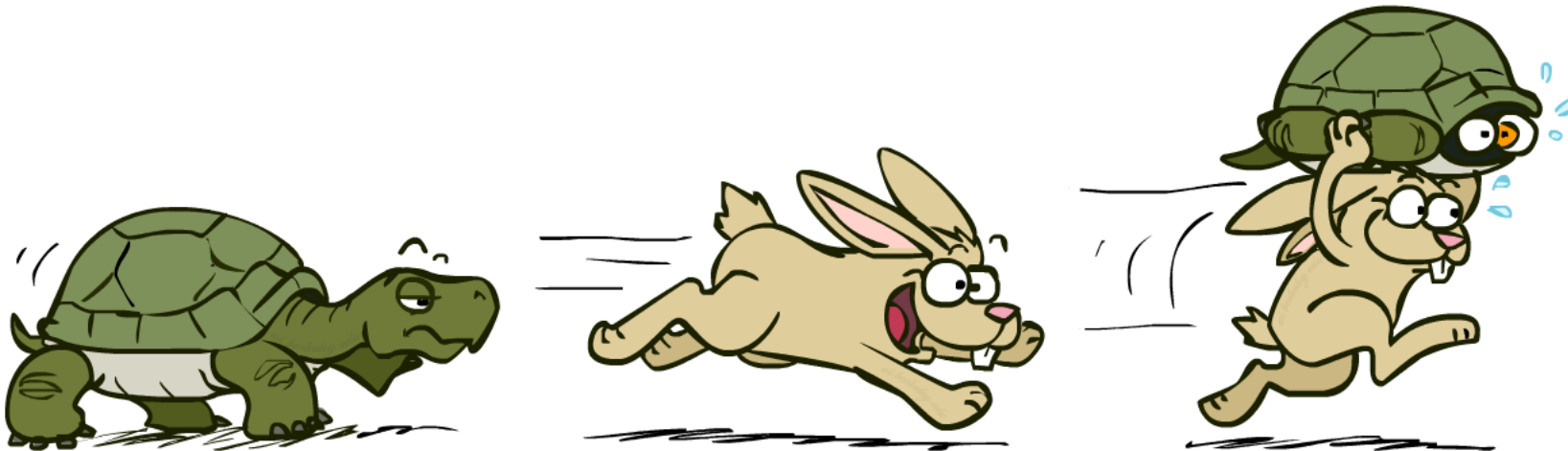
- Video games
- Pathing / routing problems
- Resource planning problems
- Robot motion planning
- Language analysis
- Machine translation
- Speech recognition
- ...





# A\* Search

- Tóm tắt A\*
  - A\* sử dụng cả chi phí ngược và (ước tính) chi phí xuôi
  - A\* là tối ưu với các phương pháp heuristic có thể chấp nhận được/nhất quán
  - Thiết kế heuristic là chìa khóa



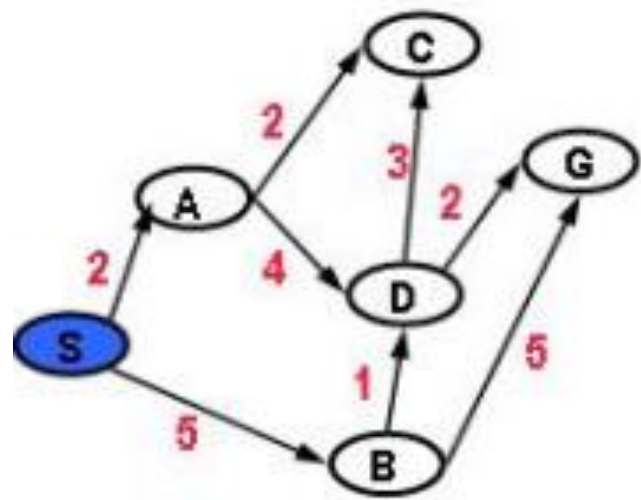
# IDA\* Search

- Ý TƯỞNG

- Nhược điểm chính của các thuật toán Greedy được trình bày là độ phức tạp về không gian của chúng.
- Ý tưởng: sử dụng các khái niệm về DFS lặp lại-làm sâu
  - Đặt ra các ngưỡng  $0, \alpha, 2\alpha, 3\alpha, \dots$
  - Không mở rộng nút có hàm  $f > 0$ , nếu tìm được đích thì dừng lại.
  - Không mở rộng nút có hàm  $f > \alpha$ , nếu tìm được đích thì dừng lại.
  - Không mở rộng nút có hàm  $f > 2\alpha$ , nếu tìm được đích thì dừng lại.
  - .....

# IDA\* Search

Ví dụ:



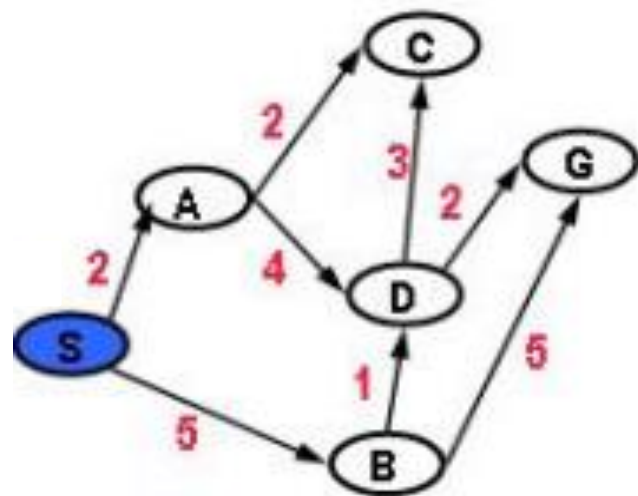
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	

# IDA\* Search

Ví dụ:



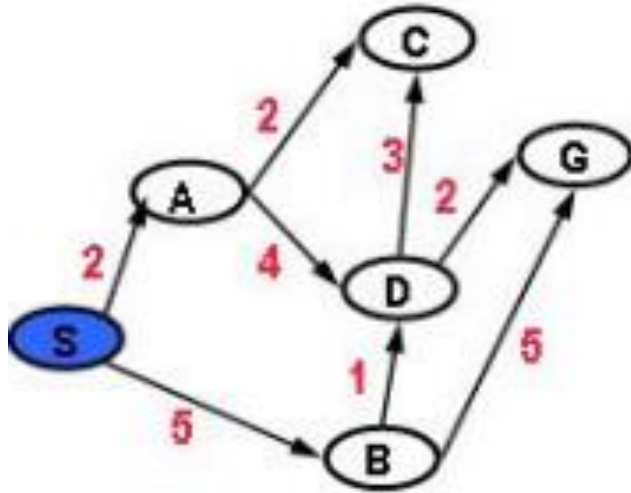
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>

# IDA\* Search

Ví dụ:



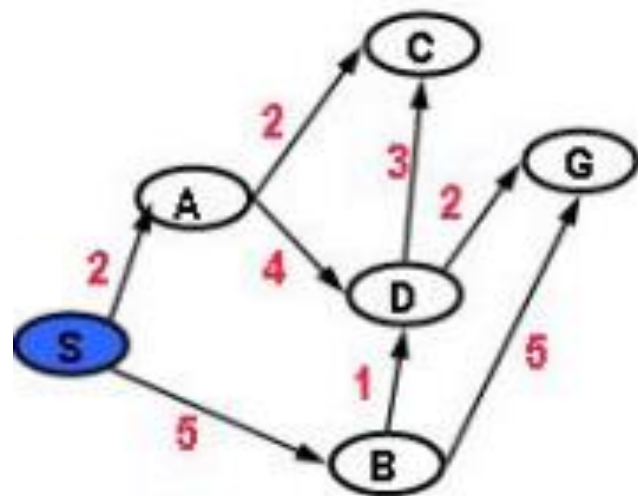
### Heuristic Values

A=2	C=1	S=0
B=3	D=1	G=0

[illegible]

# IDA\* Search

Ví dụ:



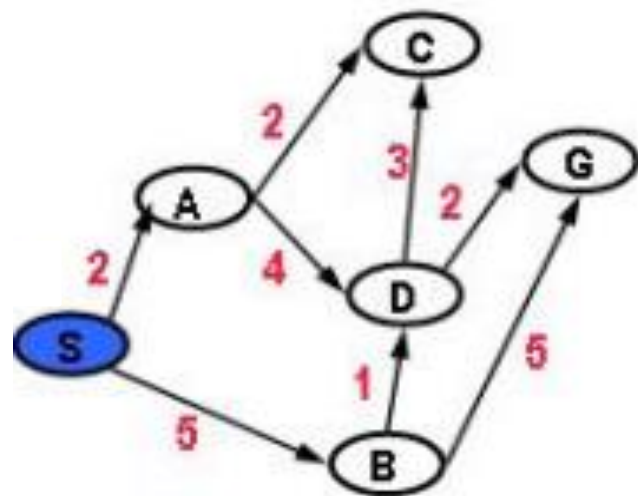
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>

# IDA\* Search

Ví dụ:



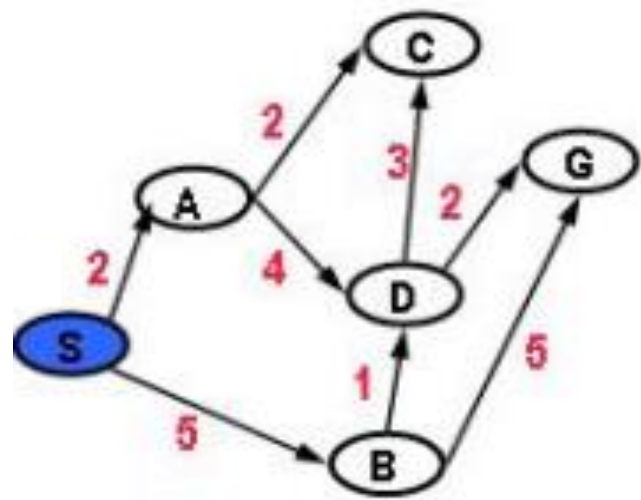
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	

# IDA\* Search

Ví dụ:



Heuristic Values

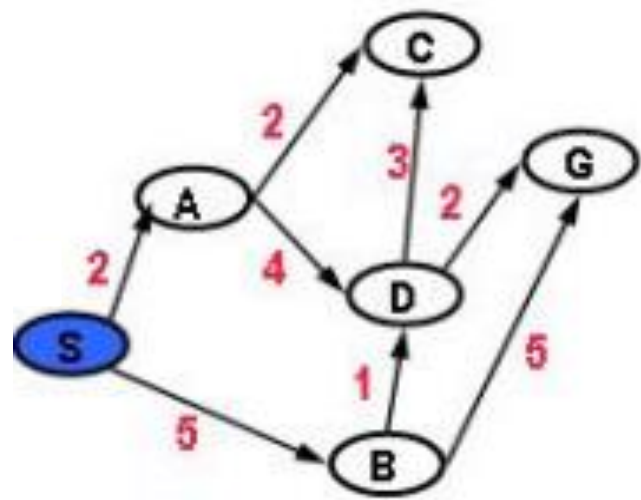
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>



# IDA\* Search

Ví dụ:



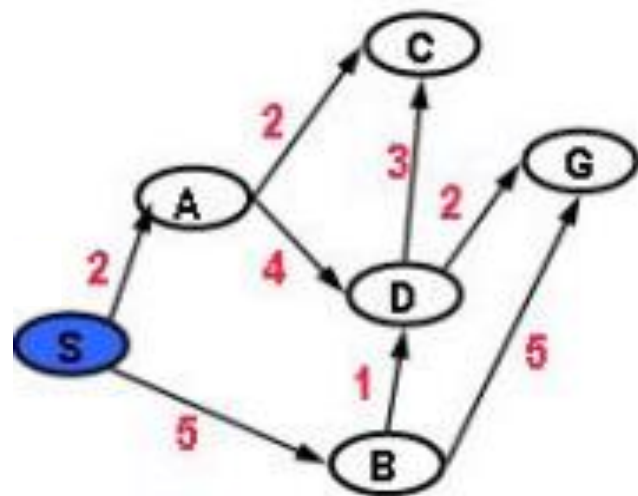
Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	

# IDA\* Search

Ví dụ:



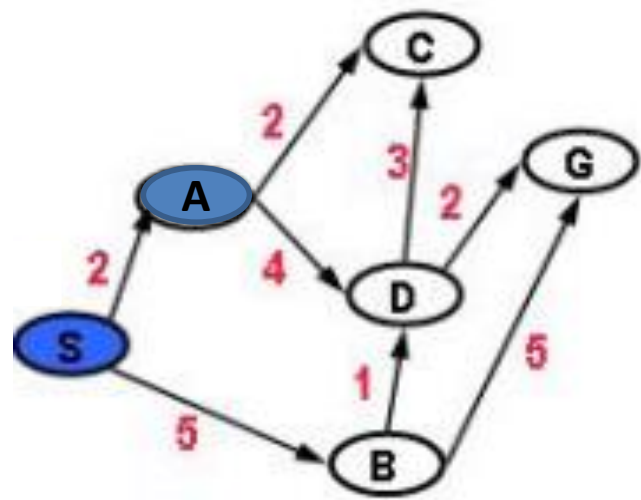
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>

# IDA\* Search

Ví dụ:



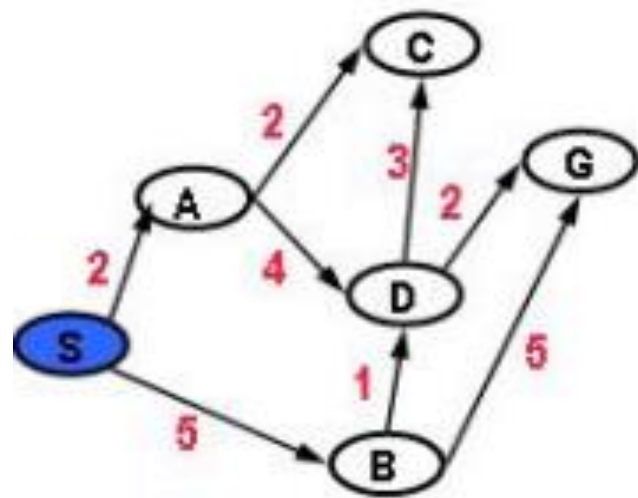
Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	

# IDA\* Search

Ví dụ:



Heuristic Values

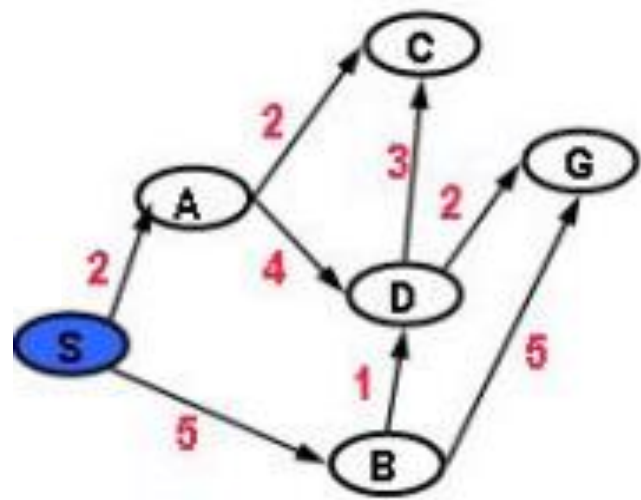
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	

# IDA\* Search

Ví dụ:



Heuristic Values

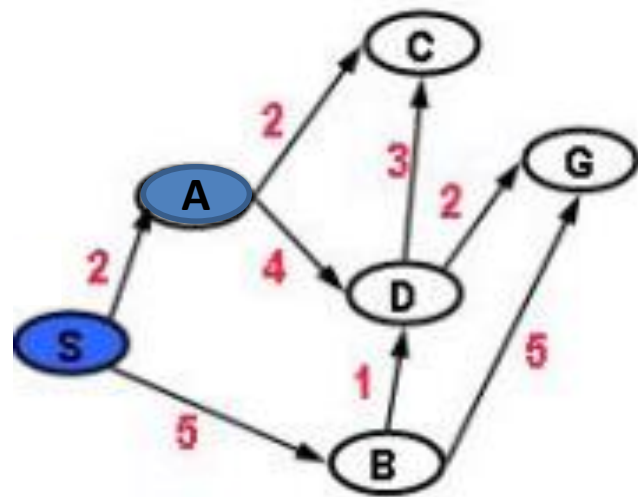
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

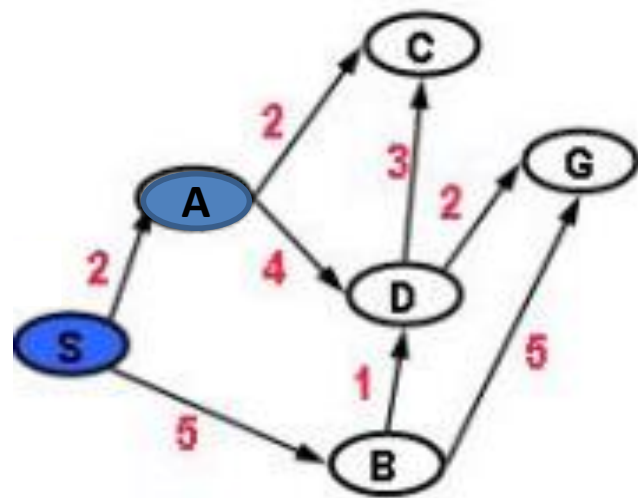
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	

# IDA\* Search

Ví dụ:



Heuristic Values

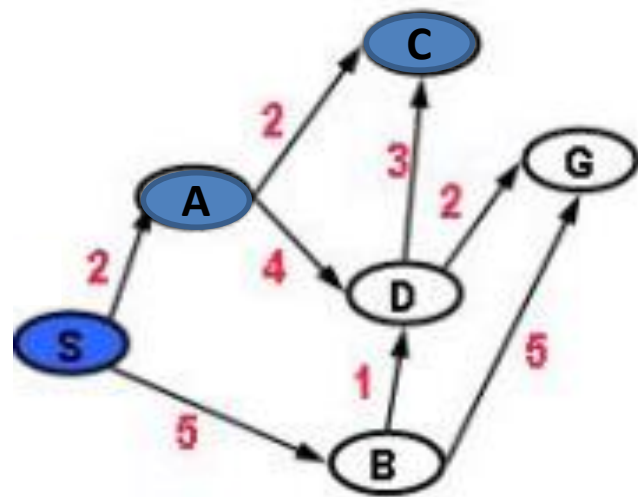
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

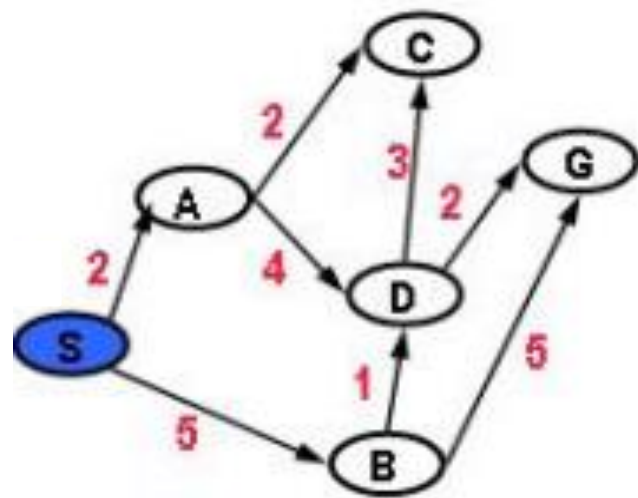
Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	



# IDA\* Search

Ví dụ:



Heuristic Values

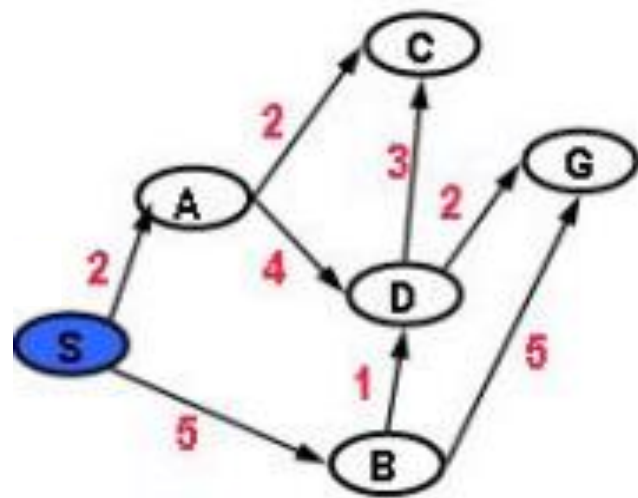
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

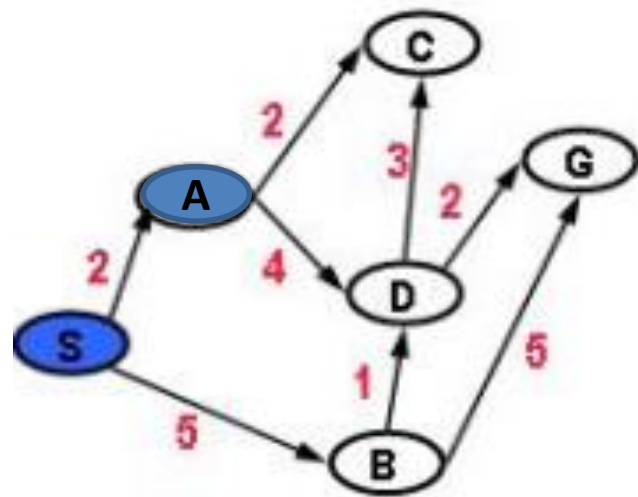
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	

# IDA\* Search

Ví dụ:



Heuristic Values

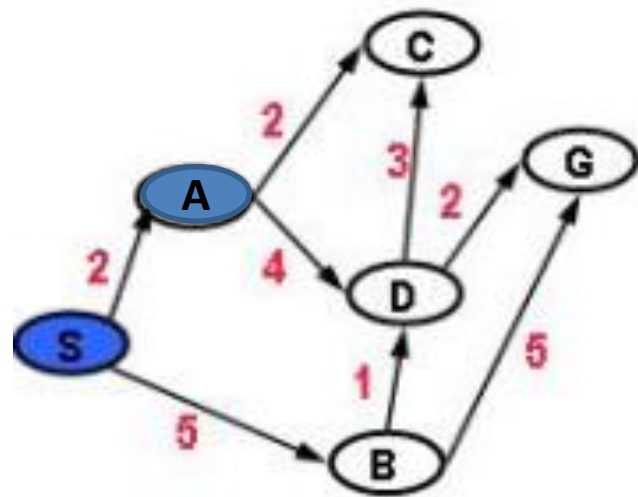
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	

# IDA\* Search

Ví dụ:



Heuristic Values

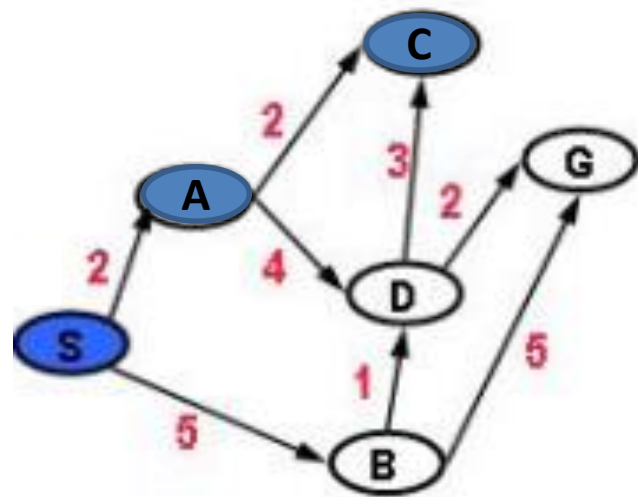
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

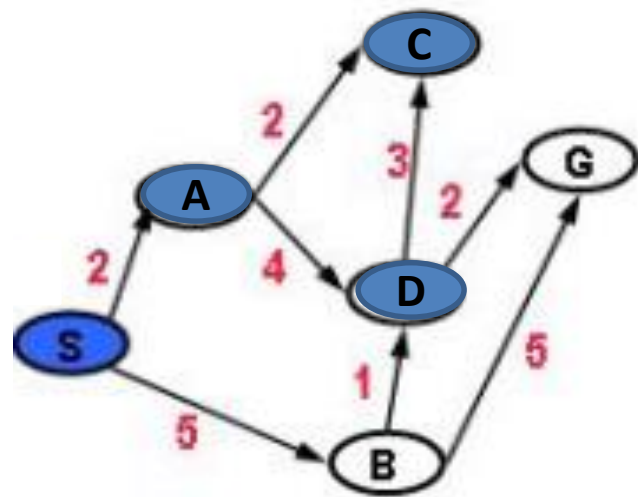
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	

# IDA\* Search

Ví dụ:



Heuristic Values

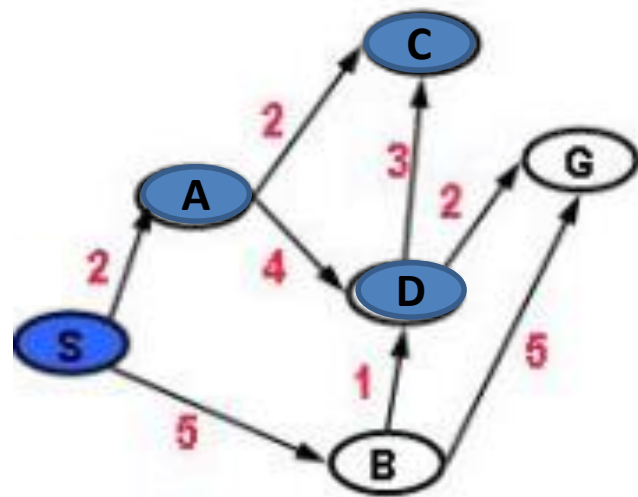
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	<u>(8, BS)</u> <u>(7, DAS)</u>
<u>(7, DAS)</u>	

# IDA\* Search

Ví dụ:



Heuristic Values

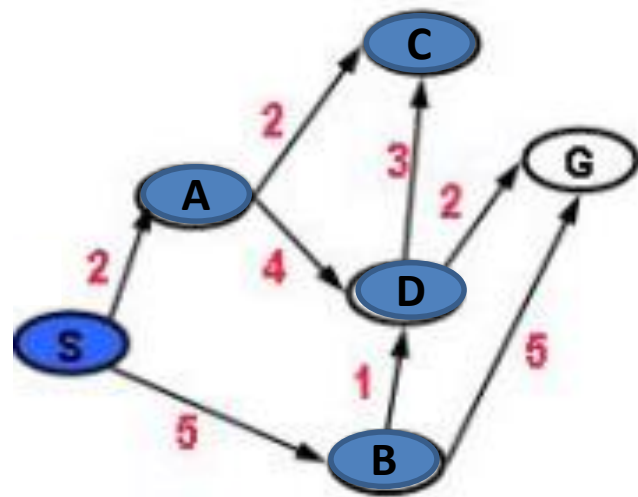
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	<u>(8, BS)</u> <u>(7, DAS)</u>
<u>(7, DAS)</u>	<u>(8, BS)</u> <u>(8, GDAS)</u> <u>(8, 10, CDAS)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

A=2      C=1      S=0  
B=3      D=1      G=0

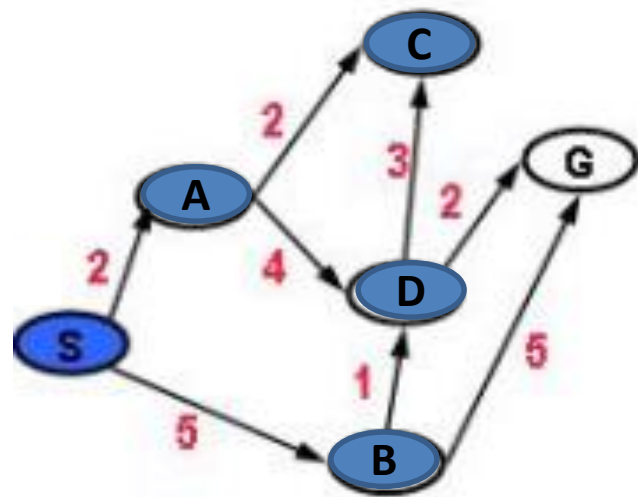
Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	<u>(8, BS)</u> <u>(7, DAS)</u>
<u>(7, DAS)</u>	<u>(8, BS)</u> <u>(8, GDAS)</u> <u>(10, CDAS)</u>
(8, BS)	



# IDA\* Search

Ví dụ:



Heuristic Values

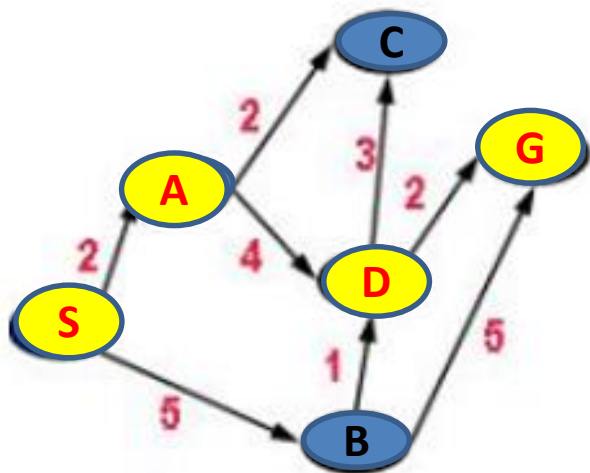
A=2      C=1      S=0  
B=3      D=1      G=0

Node hiện tại	Tập biên O
i=0	
	<u>(0, S)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	<u>(5, CAS)</u>
(5, CAS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> <u>(8, BS)</u>
(4, AS)	<u>(8, BS)</u> <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	<u>(8, BS)</u> <u>(7, DAS)</u>
<u>(7, DAS)</u>	<u>(8, BS)</u> <u>(8, GDAS)</u> <u>(10, CDAS)</u>
(8, BS)	<u>(8, GDAS)</u> <u>(8, 10, CDAS)</u> <u>(10, GBS)</u> , <u>(7, DBS)</u>

# IDA\* Search

Ví dụ:



Heuristic Values

A=2    C=1    S=0  
B=3    D=1    G=0

Node hiện tại	Tập biên O
i=6	
	<u>(0, S)</u>
i=0	
(0, S)	<u>(4, AS)</u>
	<u>(4, AS)</u>
	<u>(5, CAS)</u>
(0, S)	
i=2	
	<u>(0, S)</u>
(0, S)	
i=4	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=6	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u>
(4, AS)	
i=8	
	<u>(0, S)</u>
(0, S)	<u>(4, AS)</u> (8, BS)
(4, AS)	(8, BS) <u>(5, CAS)</u> <u>(7, DAS)</u>
(5, CAS)	(8, BS) (7, DAS)
(7, DAS)	(8, BS) <u>(8, GDAS)</u> <u>(8, 10, CDAS)</u>
(8, BS)	(8, GDAS) (10, CDAS) <u>(10, GBS)</u> , <u>(7, DBS)</u>
(8, GDAS)	<u>ĐÍCH</u>

# IDA\* Search

IDA\*(O, S, G, P, h,  $\alpha$ )

- Đầu vào: bài toán tìm kiếm, hàm heuristic h, bước nhảy  $\alpha$  cho ngưỡng
  - Đầu ra: đường đi ngắn nhất từ nút xuất phát đến nút đích
  - Khởi tạo: giá trị  $i=0$  là ngưỡng cho hàm f
- 

While (i) do

$O \leftarrow S$

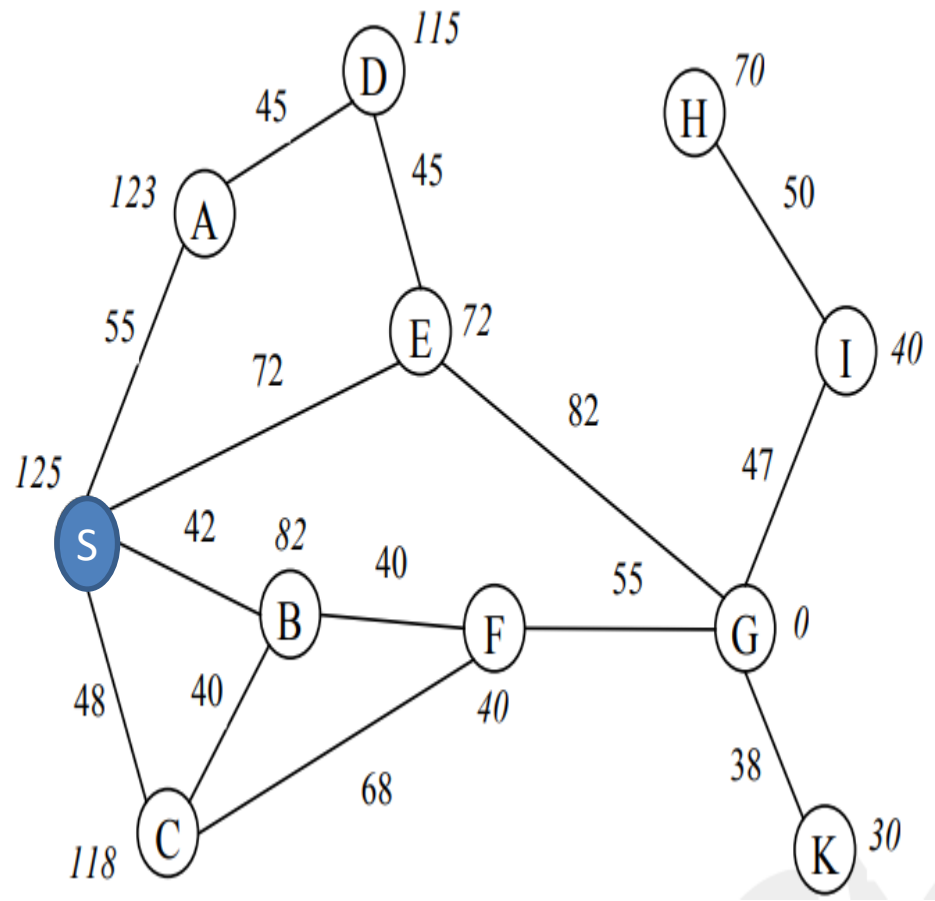
    While(O không rỗng) do

1. Lấy nút N có  $f(N)$  nhỏ nhất ra khỏi O
2. Nếu n thuộc G, return(đường đi tới N)
3. Với mọi  $M \in P(N)$ 
  - i.  $g(M) = g(N) + c(M,N)$
  - ii.  $f(M) = g(M) + h(M)$
  - iii. if ( $f(M) \leq i$ ) then thêm M vào O cùng giá trị  $f(M)$

$i = i + \alpha$

# IDA\* Search

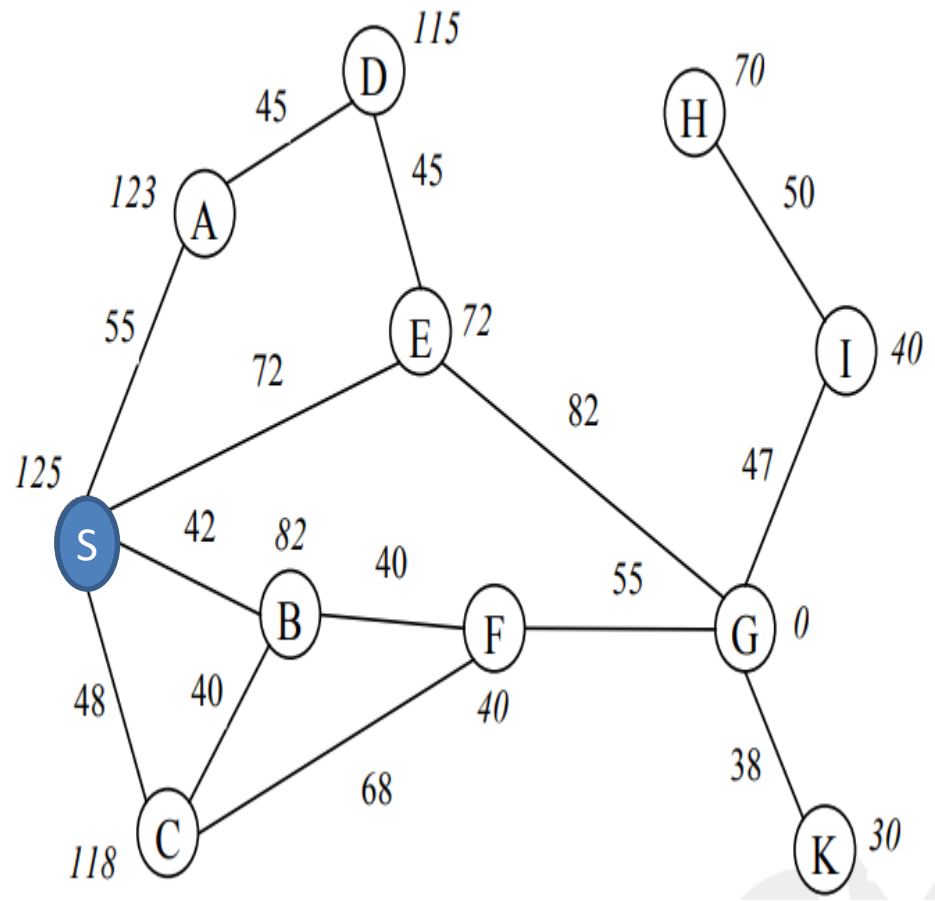
Ví dụ:



Node được xét	Tập biên O
i=0	

# IDA\* Search

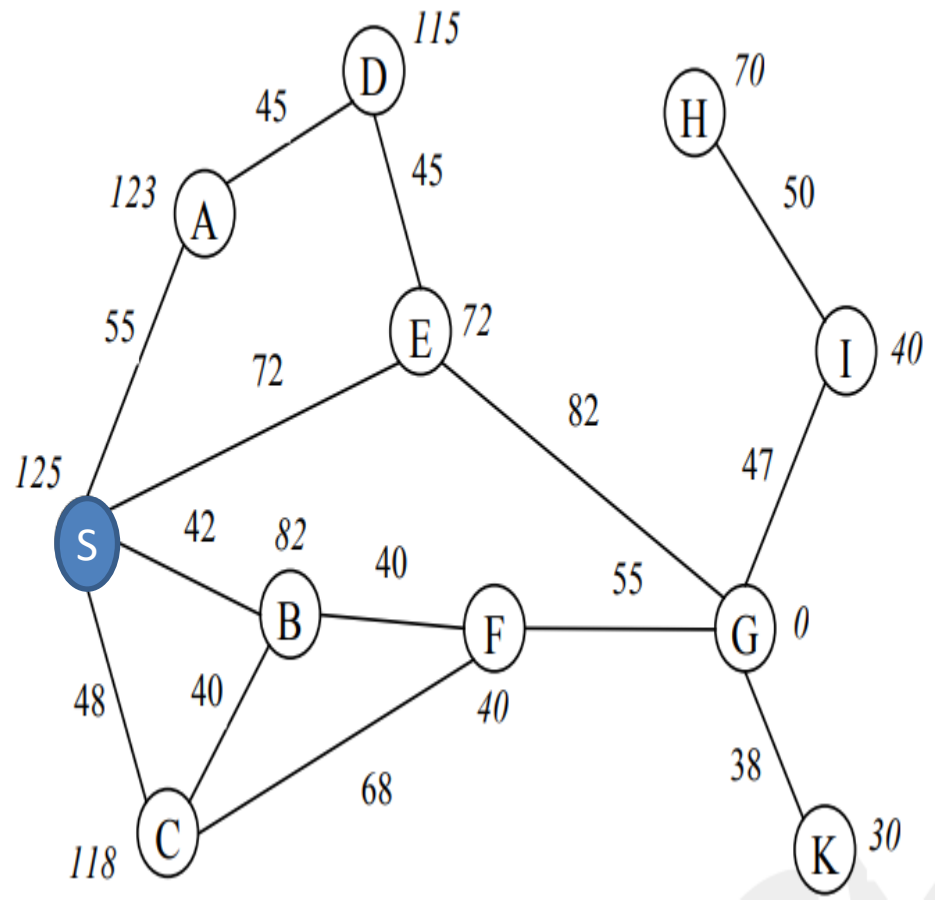
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>

# IDA\* Search

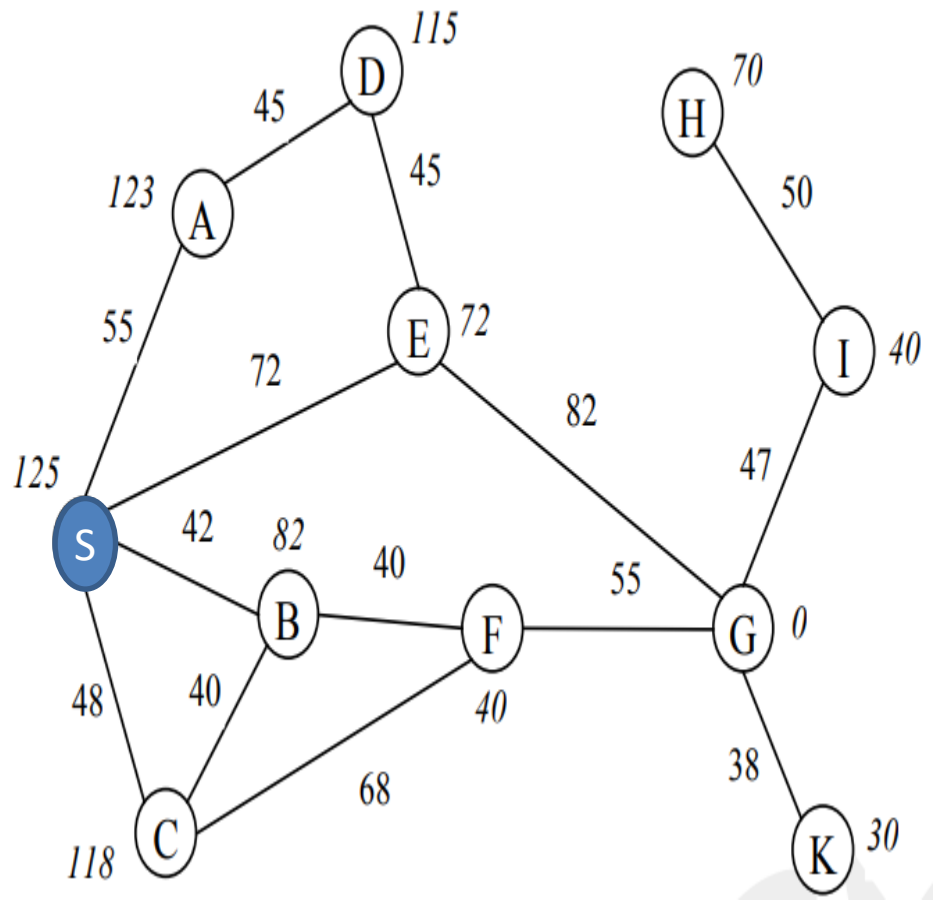
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	

# IDA\* Search

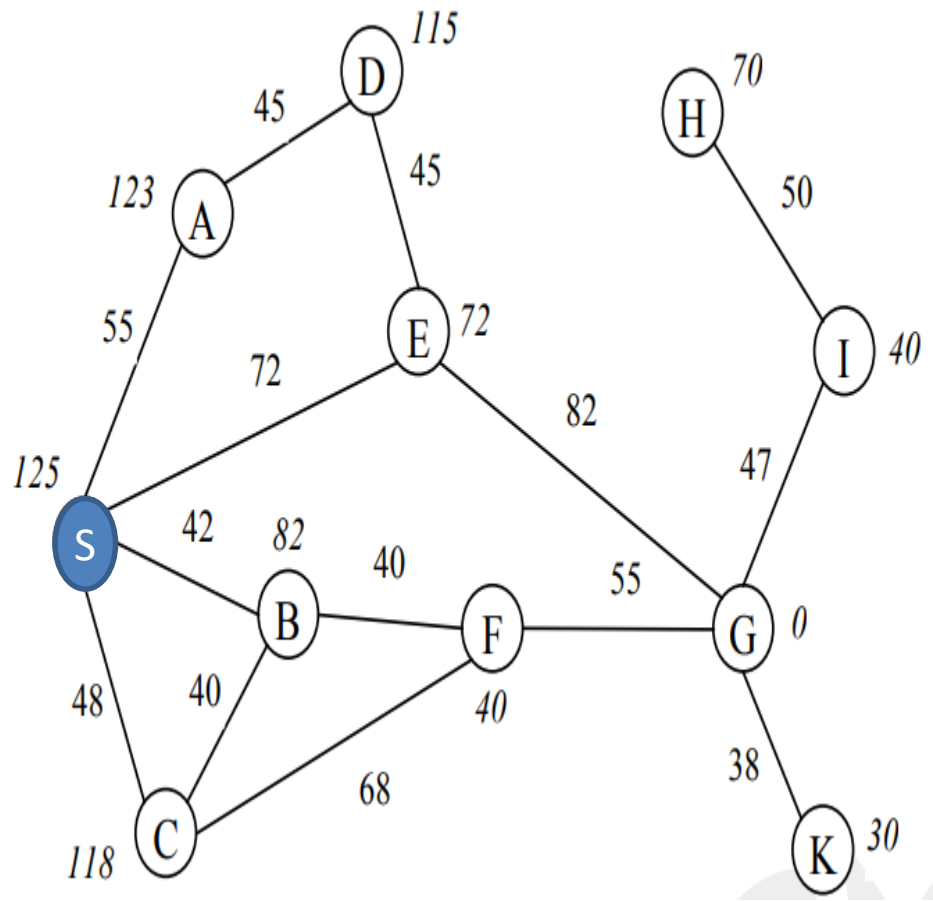
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	

# IDA\* Search

Ví dụ:

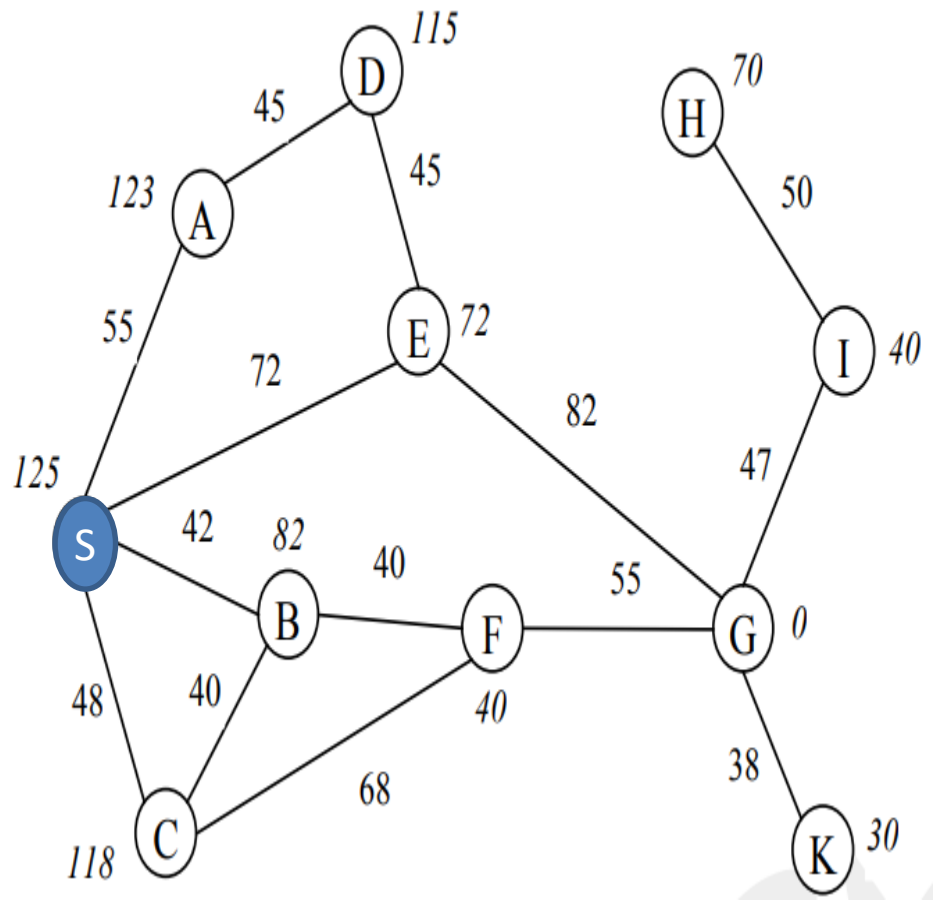


Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>



# IDA\* Search

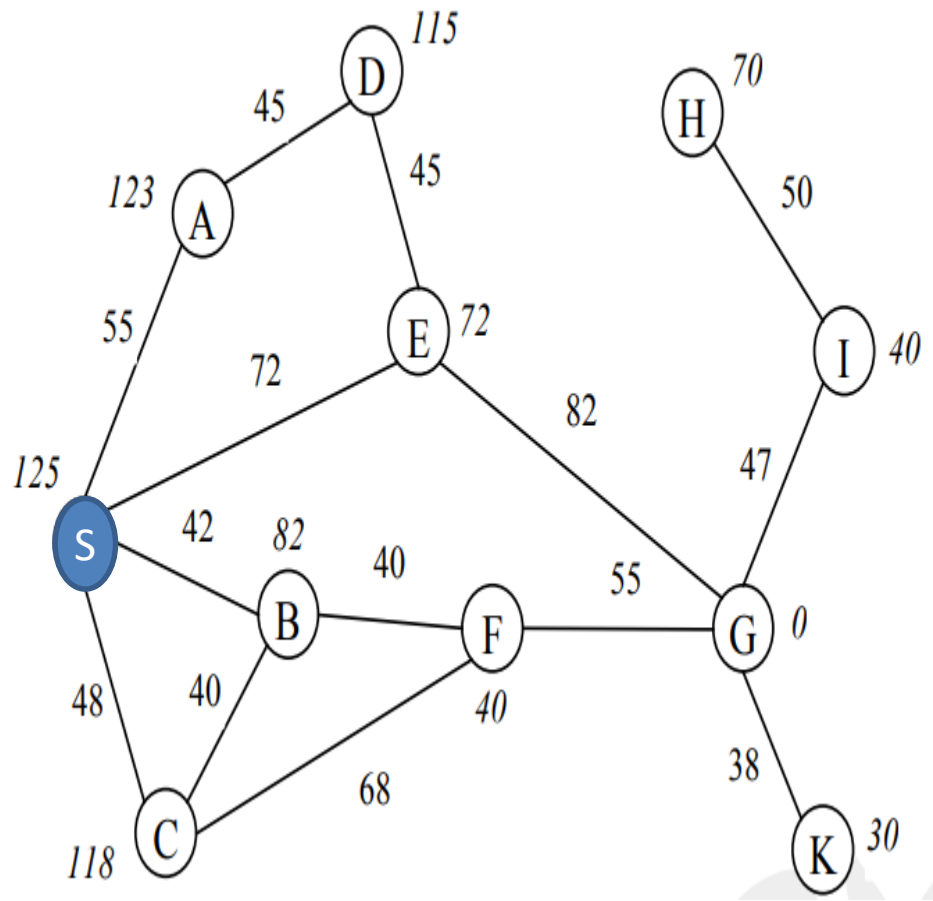
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	

# IDA\* Search

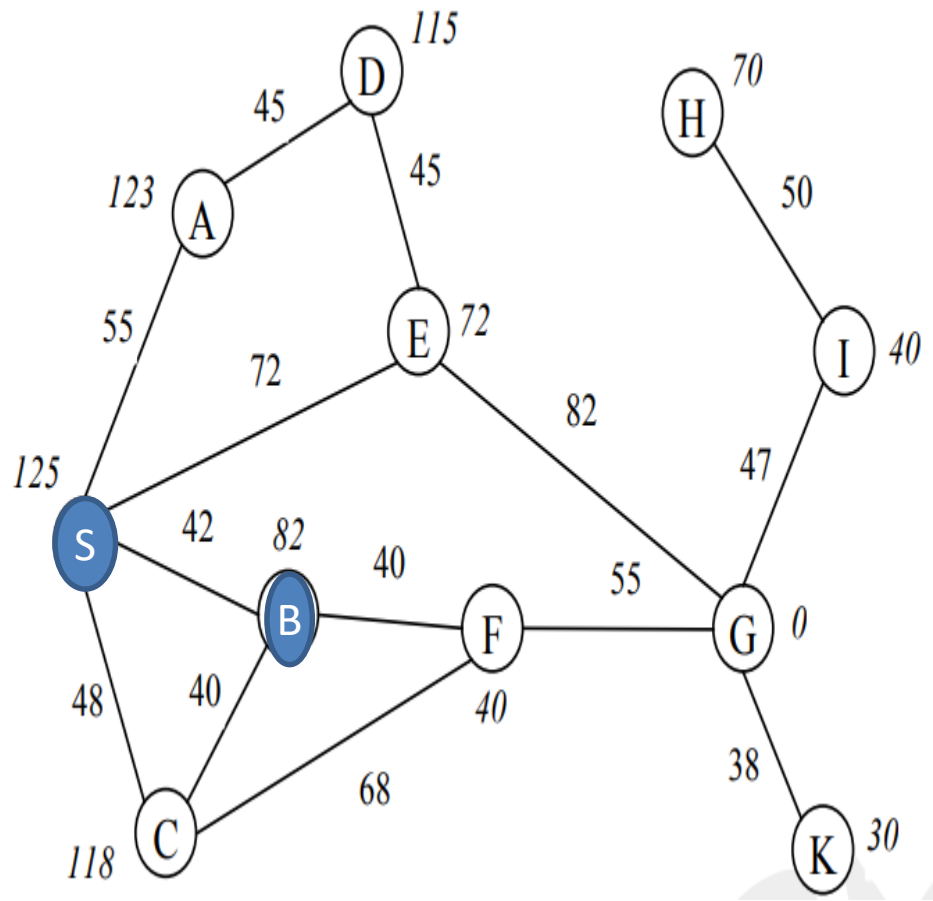
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> <u>(124,BS)</u> <u>(166,CS)</u> <u>(144,ES)</u>

# IDA\* Search

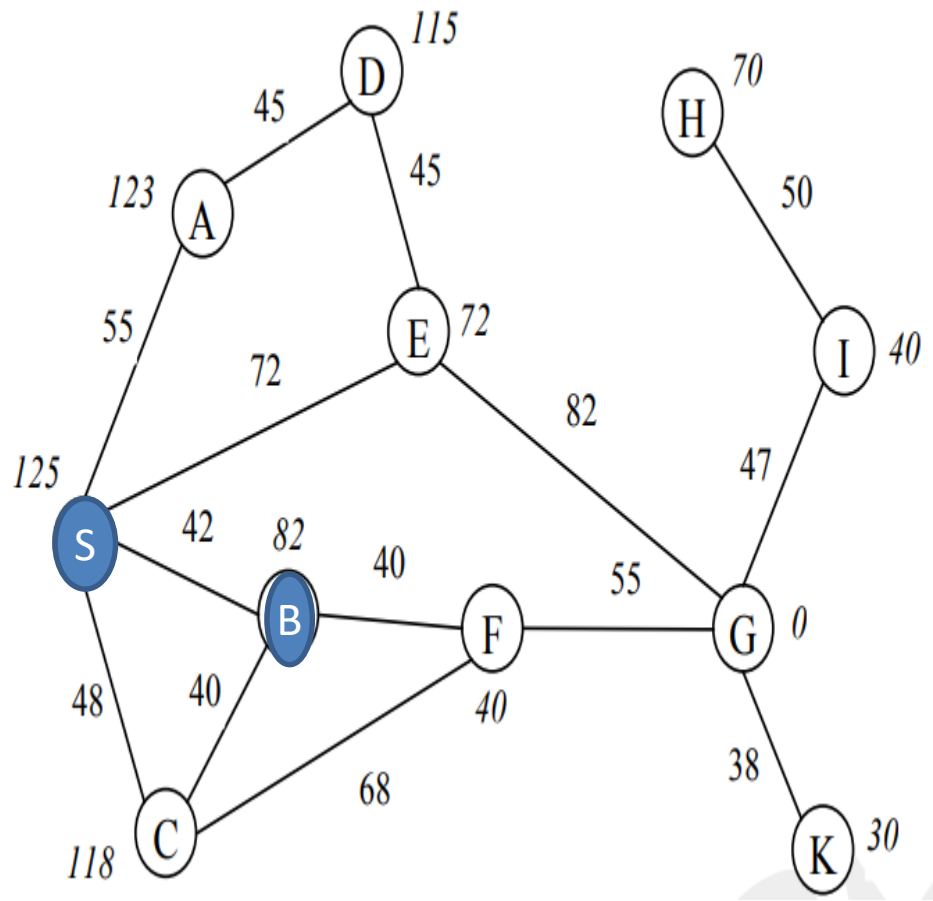
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS) (124,BS) (166,CS) (144,ES)</u>
(124,BS)	

# IDA\* Search

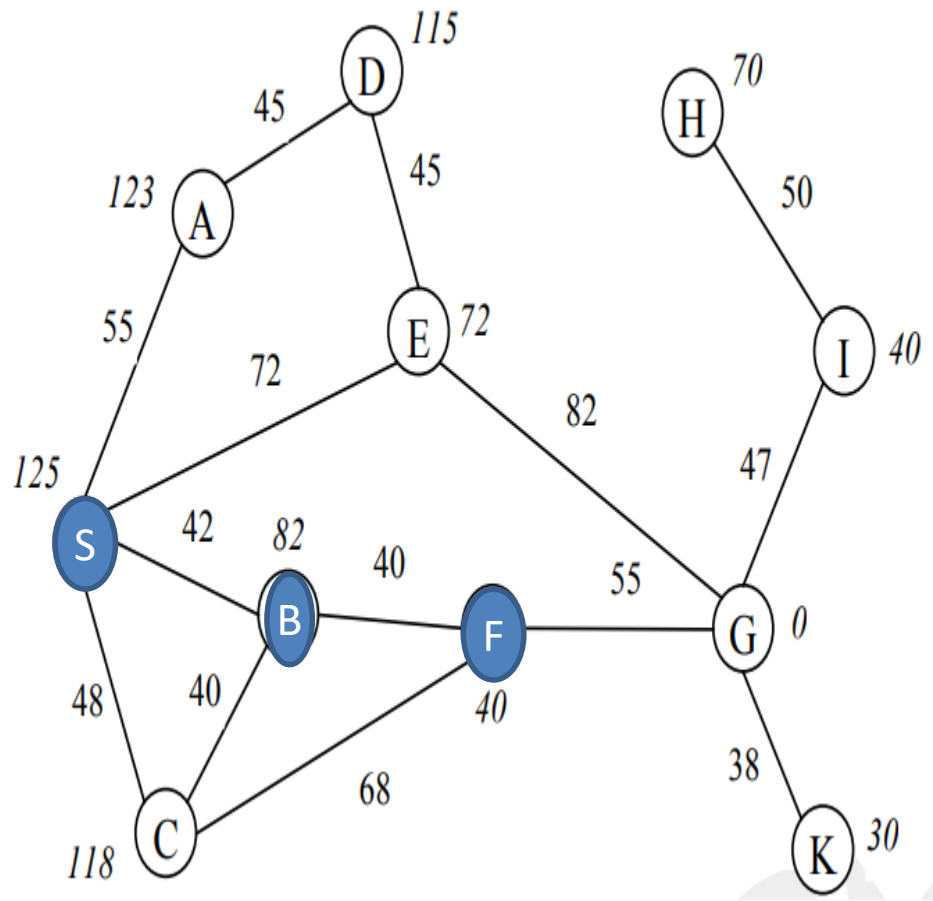
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>

# IDA\* Search

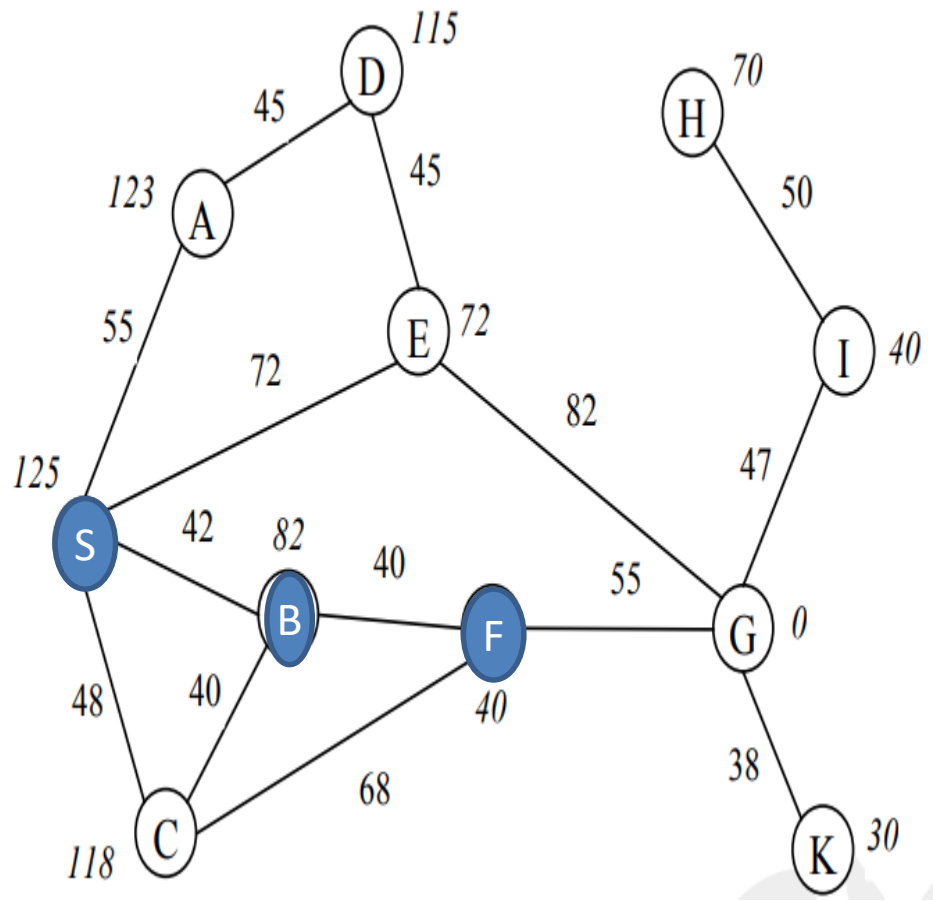
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>
(122,FBS)	

# IDA\* Search

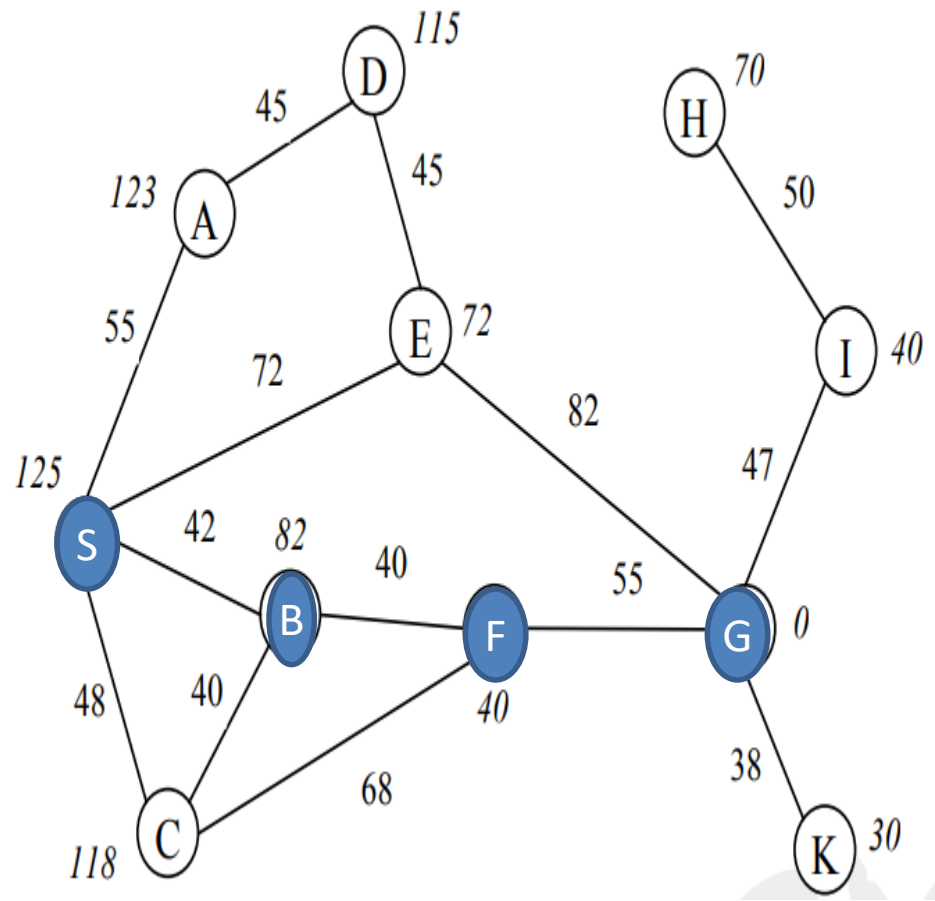
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> <u>(124,BS)</u> <u>(166,CS)</u> <u>(144,ES)</u>
(124,BS)	<u>(178,AS)</u> <u>(166,CS)</u> <u>(144,ES)</u> <u>(200,CBS)</u> <u>(122,FBS)</u>
(122,FBS)	<u>(178,AS)</u> <u>(166,CS)</u> <u>(144,ES)</u> <u>(200,CBS)</u> <u>(137,GFBS)</u> <u>(268,CFBS)</u>

# IDA\* Search

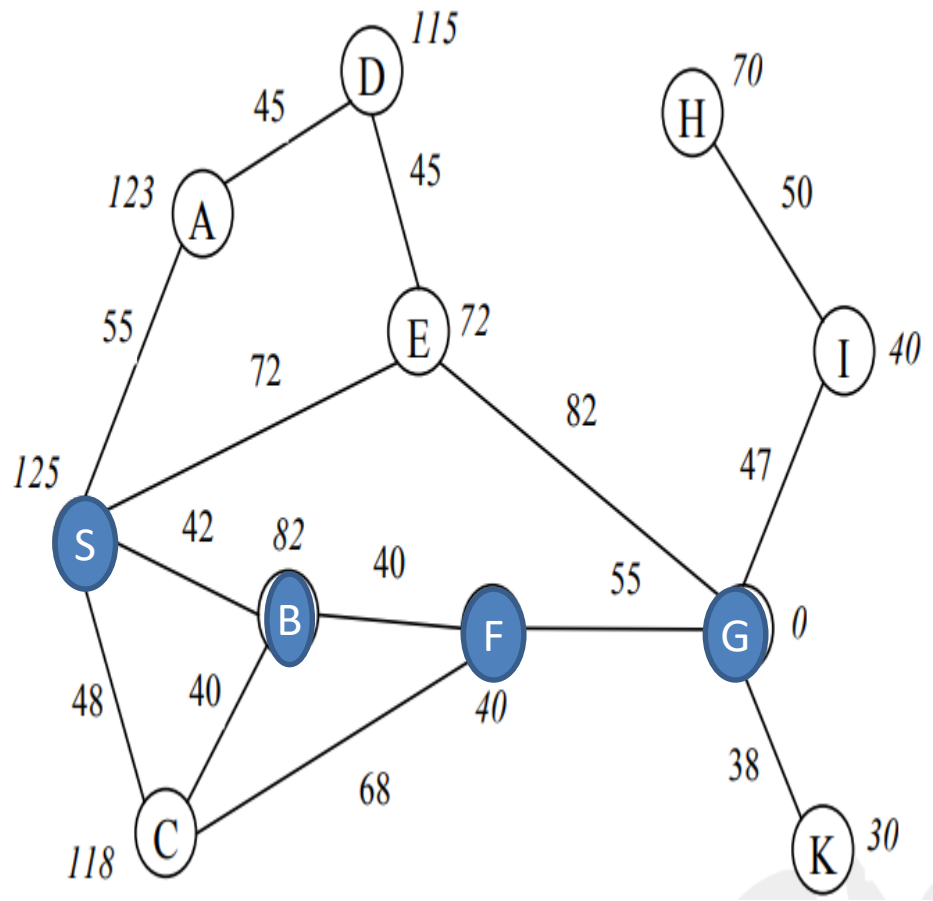
Ví dụ:



Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) <u>(137,GFBS)</u>
(137,GFBS)	

# IDA\* Search

Ví dụ:

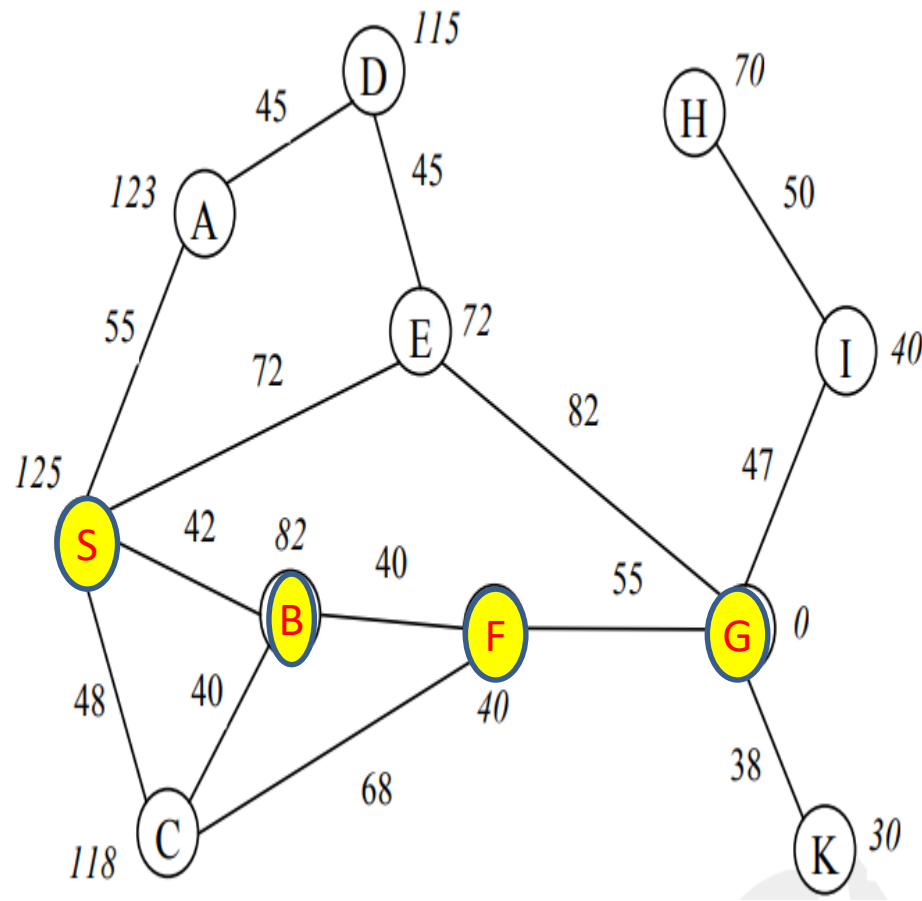


Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) <u>(137,GFBS)</u>
(137,GFBS)	<u>ĐÍCH</u>



# IDA\* Search

Ví dụ:

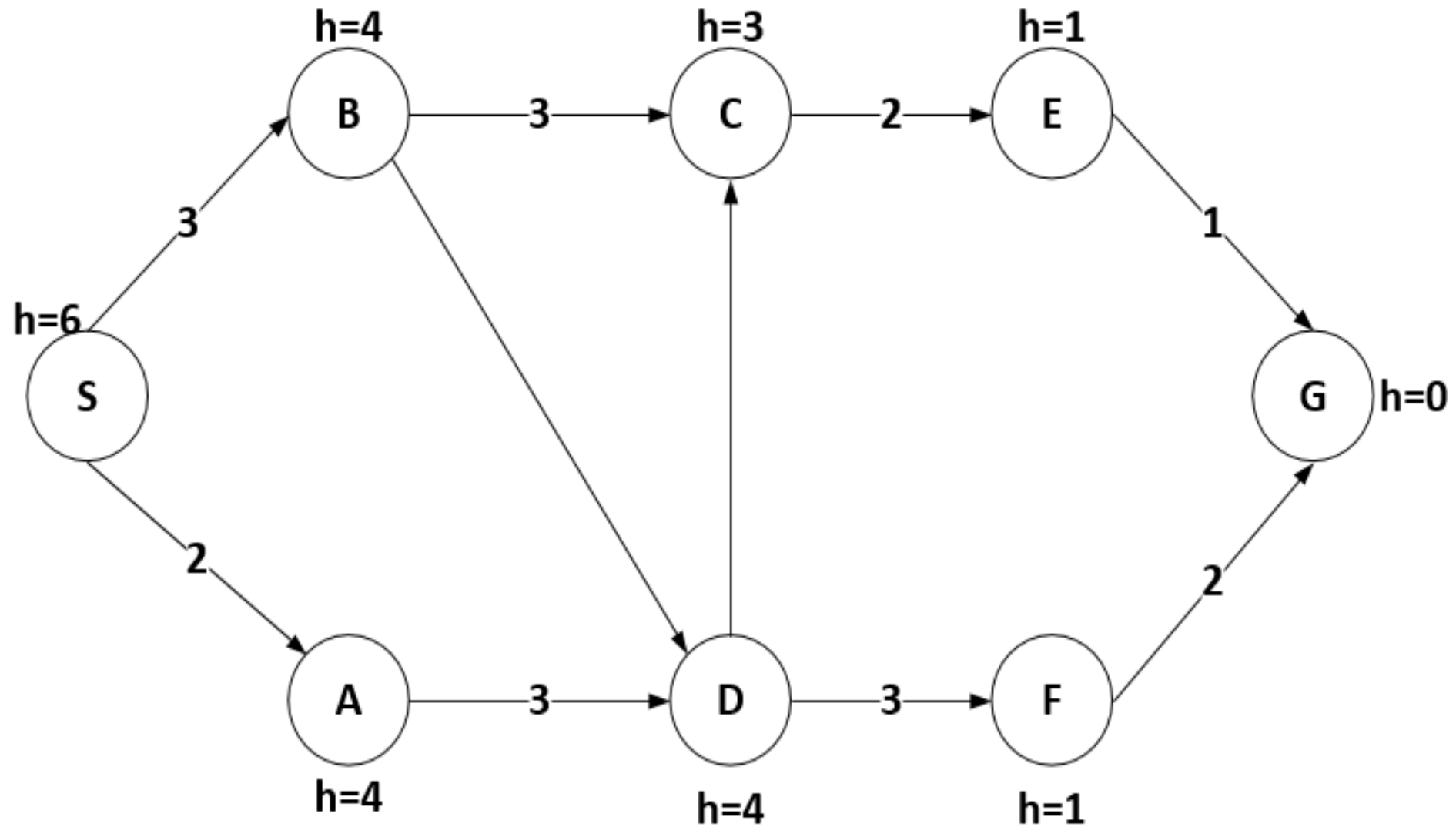


Node được xét	Tập biên O
i=0	
	<u>(125, S)</u>
(125, S)	
i=200	
	<u>(125, S)</u>
(125, S)	<u>(178,AS)</u> (124,BS) (166,CS) (144,ES)
(124,BS)	(178,AS) (166,CS) (144,ES) <u>(200,CBS)</u> <u>(122,FBS)</u>
(122,FBS)	(178,AS) (166,CS) (144,ES) (200,CBS) <u>(137,GFBS)</u>
(137,GFBS)	<u>ĐÍCH</u>

Chúng ta đã tìm ra con đường (S->B->F->G) với chi phí bằng 137

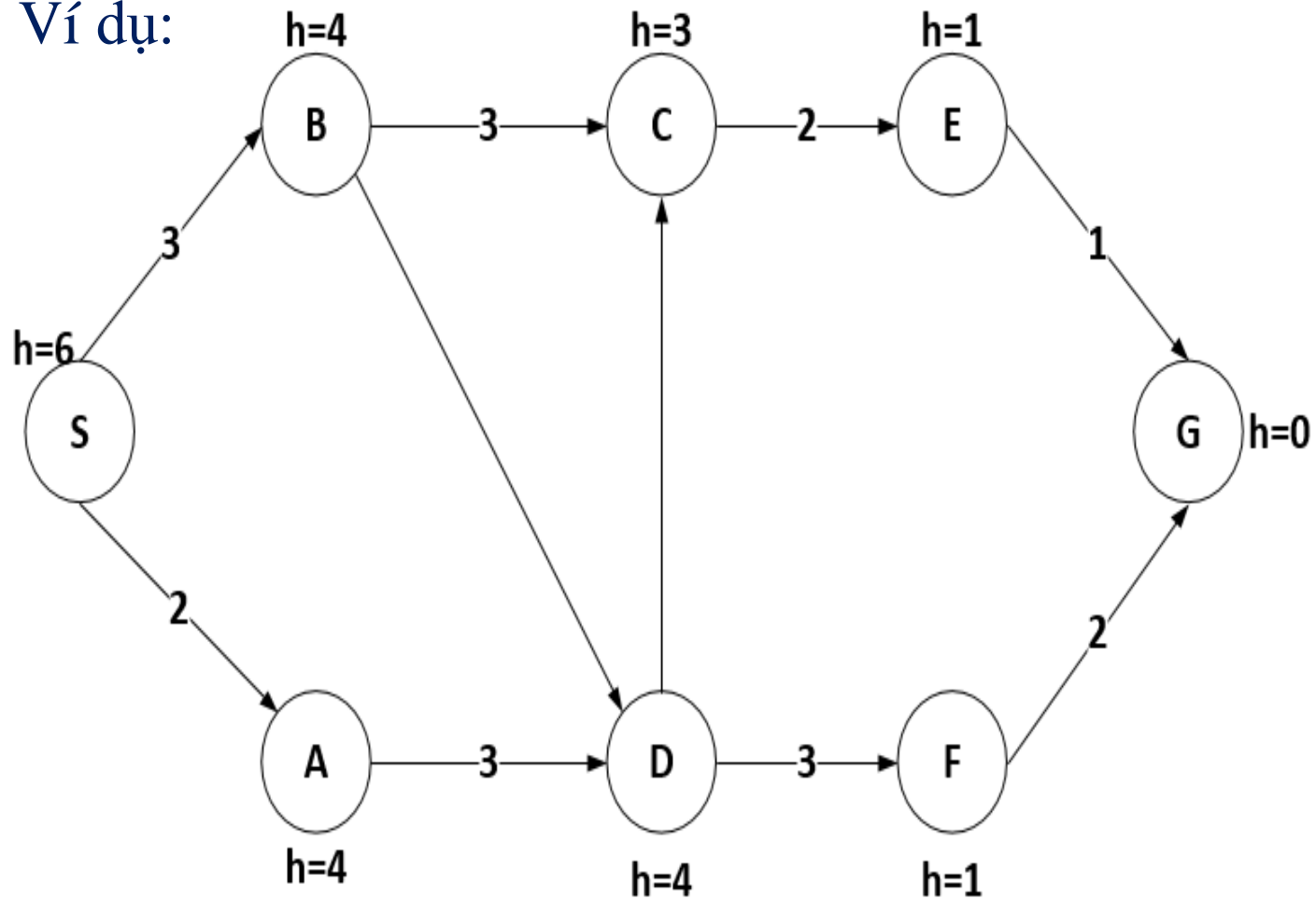
# IDA\* Search

Ví dụ:



# IDA\* Search

Ví dụ:



Node được xét	Tập biên O

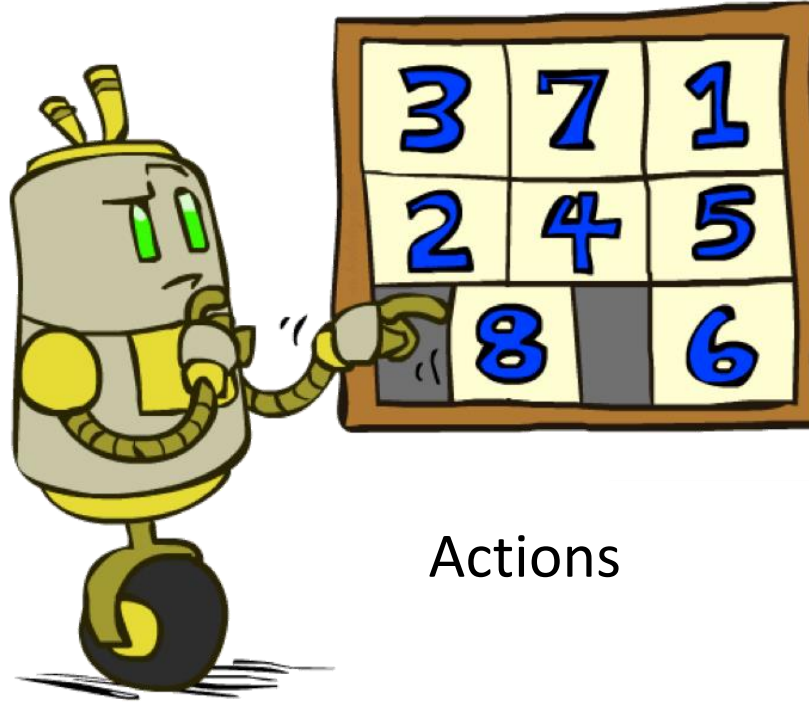
# IDA\* Search

- Tính chất của thuật toán Tìm kiếm sâu dần:
  - bán hoàn chỉnh nếu h an toàn và  $g(n) > 0$  cho tất cả các hành động n
  - tối ưu nếu H có thể chấp nhận được
  - độ phức tạp không gian  $O(lb)$ , trong đó:
    - l: chiều dài của path được tạo dài nhất
    - B: số nhánh

# Example: 8 Puzzle

7	2	4
5		6
8	3	1

Start State



Actions

	1	2
3	4	5
6	7	8

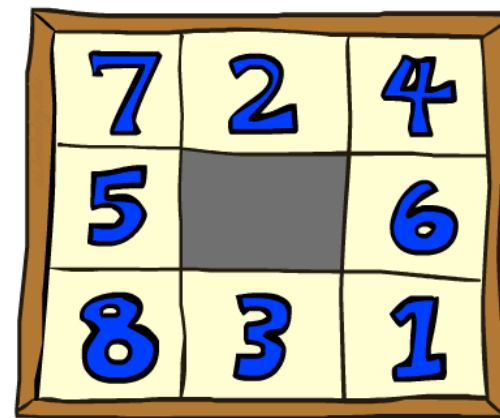
Goal State

- What are the states?
- How many states?
- What are the actions?
- How many successors from the start state?
- What should the costs be?

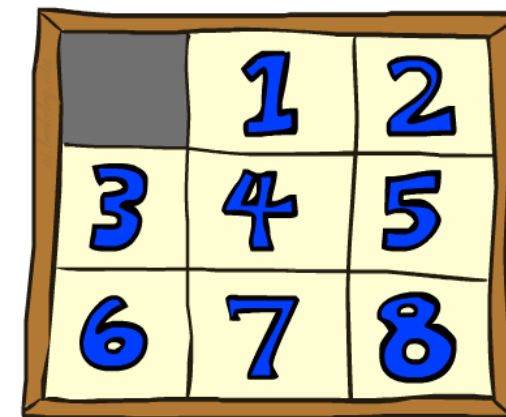
# 8 Puzzle heuristics

## 8 Puzzle I

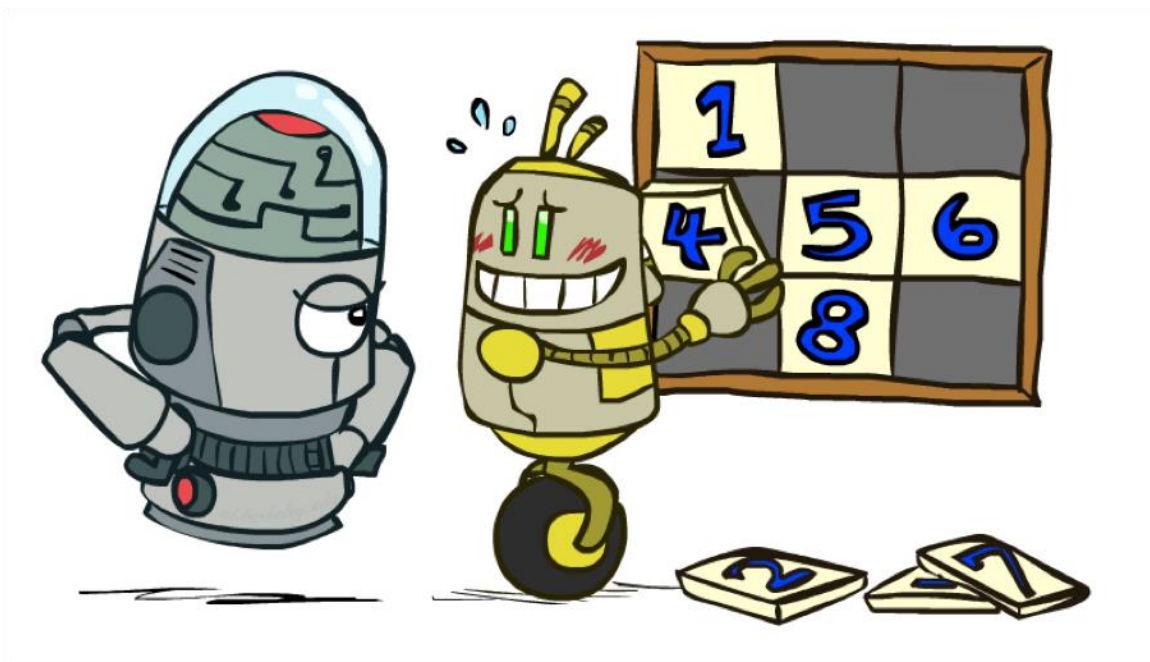
- Heuristic: Number of tiles misplaced
- Why is it admissible?
- $h_1(\text{start}) = 8$
- This is a *relaxed-problem* heuristic



Start State



Goal State



Average nodes expanded  
when the optimal path has...

	...4 steps	...8 steps	...12 steps
UCS	112	6,300	$3.6 \times 10^6$
TILES	13	39	227

# 8 Puzzle heuristics

4	6	5
7	2	
1	3	8

3	1	8
7	4	2
6	5	

admissible?

tiles misplaced

Tile	n	Goal
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
6	1	0
7	0	0
8	1	0
<b>h1(n)</b>	<b>7</b>	
<b>h1(goal)</b>	<b>0</b>	

# 8 Puzzle heuristics

4	6	5
7	2	
1	3	8

3	1	8
7	4	2
6	5	

admissible?

*Manhattan distance*

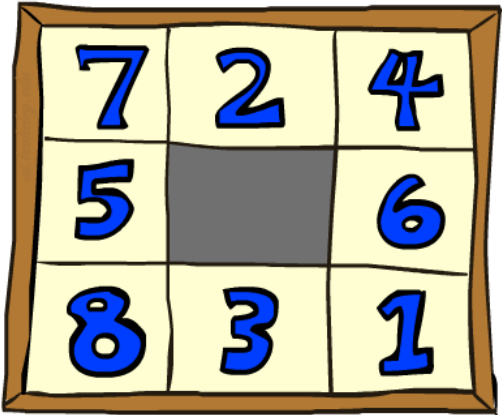
Tile	n	Goal
1	3	
2	1	
3	3	
4	2	
5	3	
6	3	
7	0	
8	2	
$h_2(n)$	17	
$h_2(goal)$	0	



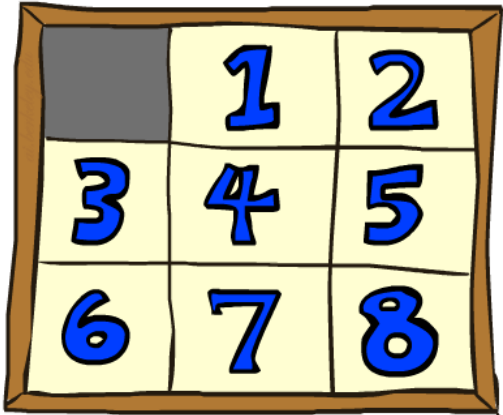
# 8 Puzzle heuristics

## 8 Puzzle II

- What if we had an easier 8-puzzle where any tile could slide any direction at any time, ignoring other tiles?
- **Total *Manhattan* distance**
- Why is it admissible?
- $h(\text{start}) = 3 + 1 + 2 + \dots = 18$



Start State



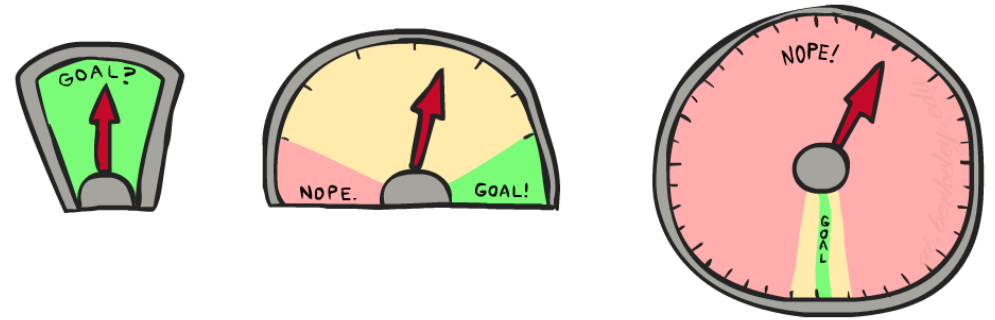
Goal State

Average nodes expanded when the optimal path has...			
	...4 steps	...8 steps	...12 steps
TILES	13	39	227
MANHATTAN	12	25	73

# 8 Puzzle heuristics

## 8 Puzzle III

- How about using the *actual cost* as a heuristic?
  - Would it be admissible?
  - Would we save on nodes expanded?
  - What's wrong with it?



- With A\*: a trade-off between quality of estimate and work per node
  - As heuristics get closer to the true cost, you will expand fewer nodes but usually do more work per node to compute the heuristic itself

# 8 Puzzle with A\*

- Solving 8 Puzzle with A\* using tree search

3	1	8
4	2	
7	6	5

start

3	1	8
7	4	2
6	5	

goal

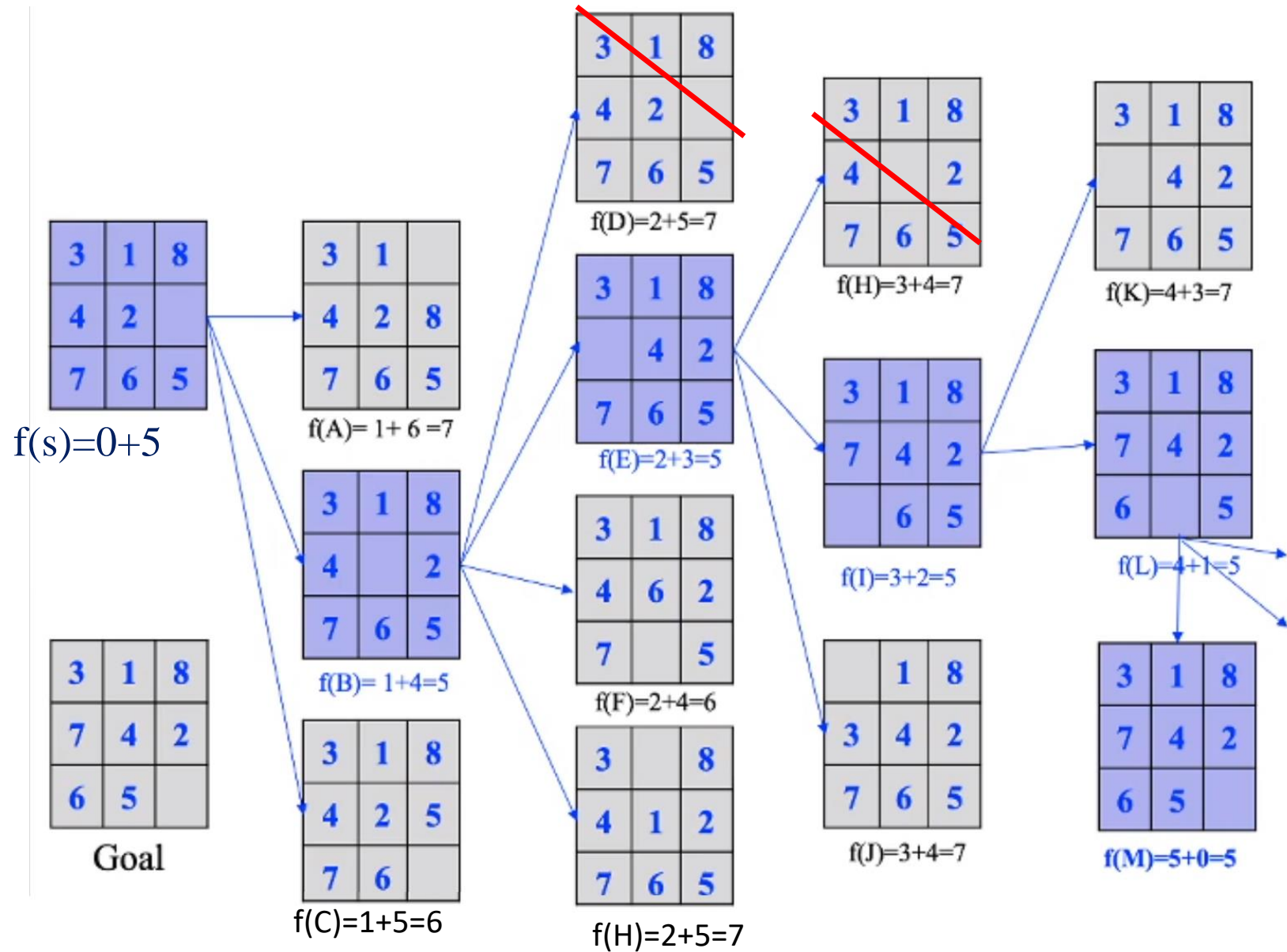
- Heuristic function  $h(n)$  = the number of misplaced tiles.
- Steps of A\*

# Application of A\*

## 8 Puzzle with A\*

- Heuristic function  $h(n)$  = the number of misplaced tiles.

A\* Search orders by the sum:  $f(n) = g(n) + h(n)$



**Thanks for your attention!**

**Q&A**

---