Ai #3

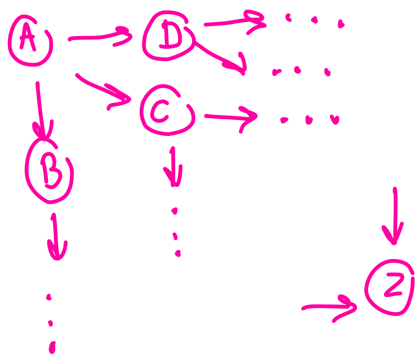
Lecture

Search Problem Formulation

Formulation

1. Initial State
2. Action Space
3. Transition function
4. Goal test function
5. Cost function

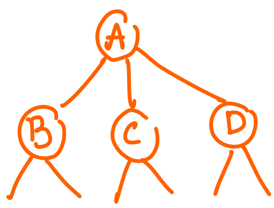
State Space



Search Problem

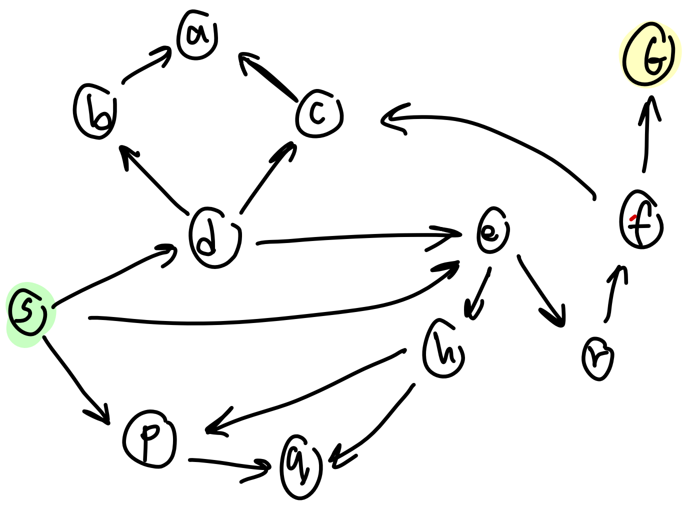
Search Algorithm

Search Tree



Uninformed-Search Strategies

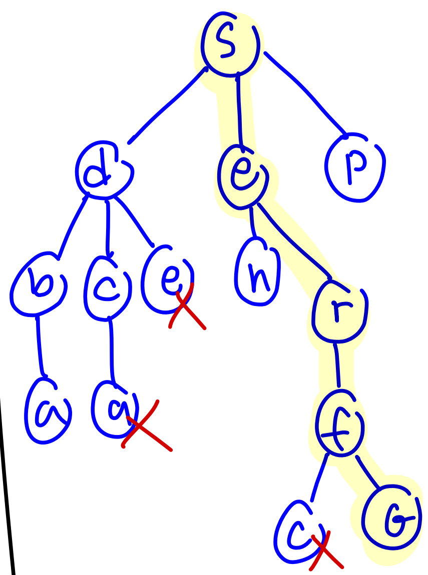
Strategy 1: Dept-First Search (DFS) "Highest-level" selected



1. Completedness : always find a plan if there is one
   - m is finite, Yes
   - (เหมาะสม) m is infinite, No
2. Optimality : minimum cost plan
   - No
3. Time complexity
   - $O(b^m)$
4. Space complexity
   - $O(bm)$

b = number of children

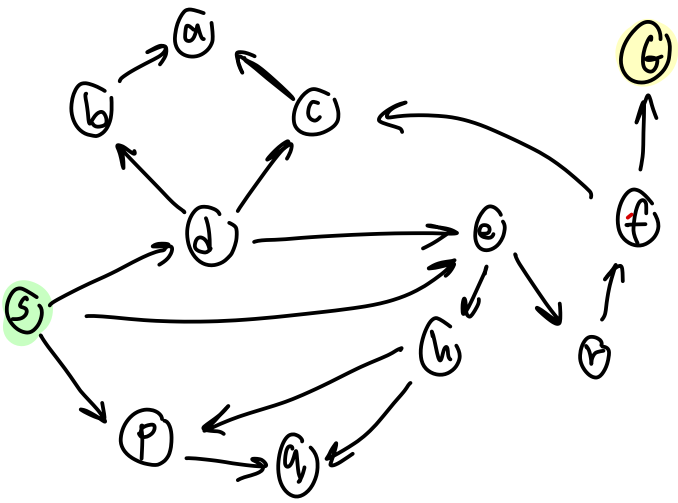
m = maximum depth

Frontier		Search Tree	Explored Set
Node	Level		State
1 S	0		S 1
2 d	1		d 2
6 e	1		b 3
p	1		a 4
3 b	2		c 5
c	2		e 6
4 a	3		
5 c	2		
r	2		
f	3		
G	4		

Plan:
S → e → r → f → G
cost: 4 steps

Strategy 2: Breadth-First Search (BFS)

"Lowest-level" selected



1. Completedness

   b is finite, Yes

   b is infinite, No

2. Optimality

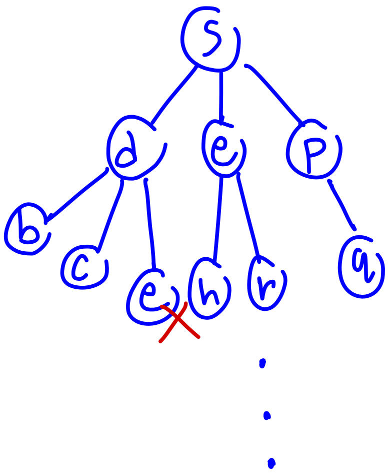
   Yes   * Shortest but not always minimum cost

3. Time complexity   s = shortest

   $O(b^s)$

4. Space complexity

   $O(b^s)$

Frontier		Search Tree	Explored Set
Node	Level		State
1 S	0		S 1
2 d	1		d 2
3 e	1		e 3
4 p	1		p 4
5 b	2		b 5
6 c	2		c 6
h	2	Plan:	. . .
r	2	s → e → r → f → g	
q	2	Cost:	
. . .	. . .	4 steps	

Strategy 3: Iterative - Deepening Search (IDS)

DFS Combining BFS

Not Optimal
Memory: $O(bm)$

Optimal (shortest)
Memory: $O(b^m)$

Iterative - Deepening Search

Round 1: Run a DFS with depth limit 1,
if no solution ... (go next Round)

Round 2: Run a DFS with depth limit 2,
if no solution ... (go next Round)

⋮

Round n: Run a DFS with depth limit n + step

~~TO DO~~

1. Completedness
   BFS: b is finite, Yes

end Lecture 3

2. Optimality
   BFS: shortest

3. Time complexity
   $O(b^m)$

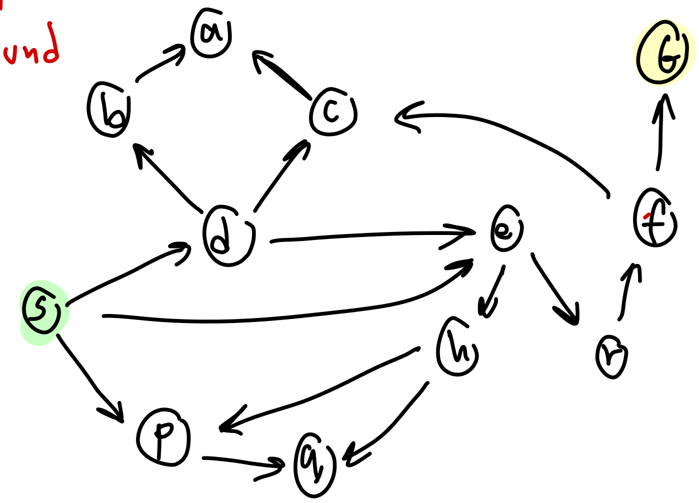
Quiz Review

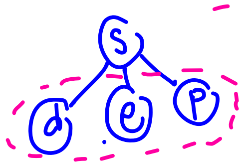
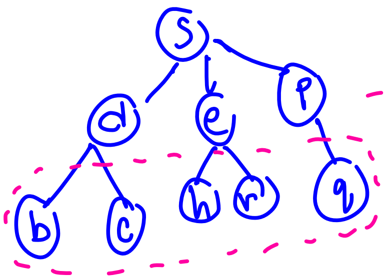
4. Space complexity
   DFS: $O(bm)$

Strategy 3: Iterative-Deepening Search (IDS)

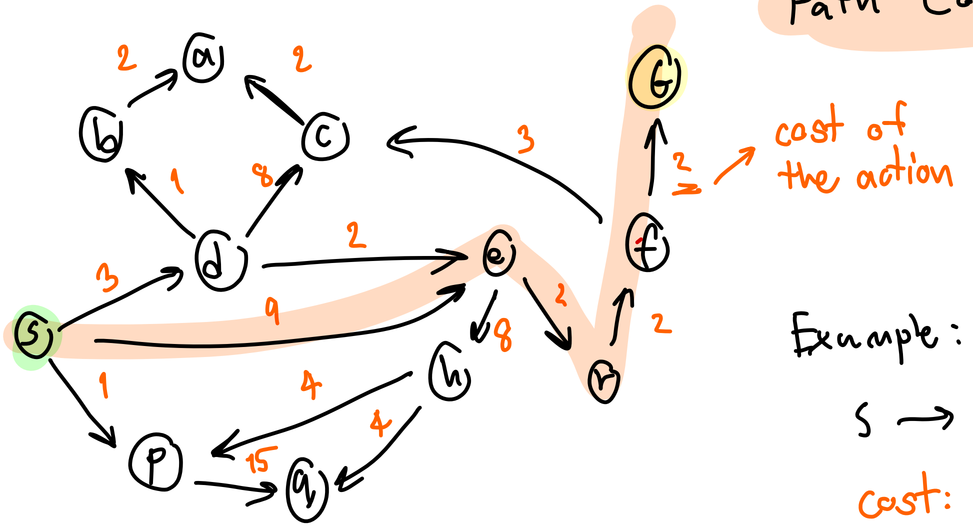
Where we need to stop?

Ans: Care Depth limit / Depth bound



Depth limit	Frontier: Node	Frontier: Level	Search Tree	Explored Set: State
0			S	S
1	S	0 zero	 - maximum depth - not a Goal	S d e p
2	d e p	1	 - maximum depth - not a Goal	S d, b, c e, h, r p, q
			∴ do it until maximum "Depth limit" it will stop	

Path Cost Function $g(n)$



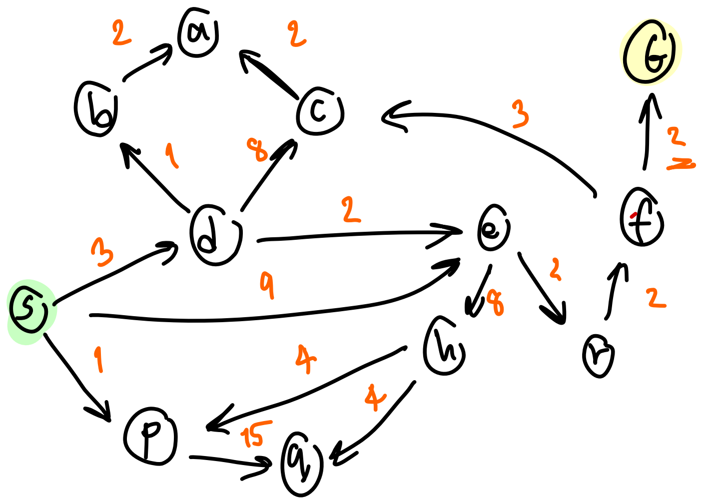
Example: BFS plan

$s \rightarrow e \rightarrow r \rightarrow f \rightarrow G$

cost: 15

Strategy 4: Uniform Cost Search (UCS)

"lowest g(n)" selected



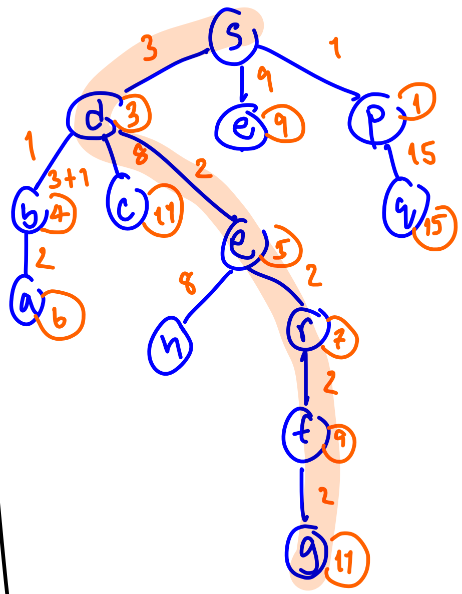
1. Completedness
   Yes
2. Optimality
   Yes (minimum cost)
3. Time complexity
   $O(b^{C^*/\varepsilon})$
4. Space complexity
   $O(b^{C^*/\varepsilon})$

C^* = shallowest step to goal
ε = each cost of cost path

$C^* = 10$
$\varepsilon = 2$

$C^*/\varepsilon = \frac{10}{2} = 5$

Frontier		Search Tree	Explored Set
Node	g(n)		State
s_1	0		s_1
d_2	3		p_2
e_3	9		d_3
p_4	1		b_4
q_5	15		c_5
e	3+2=5		e_6

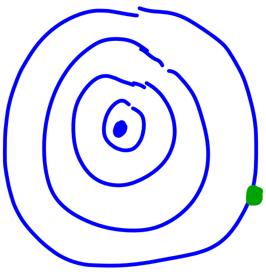


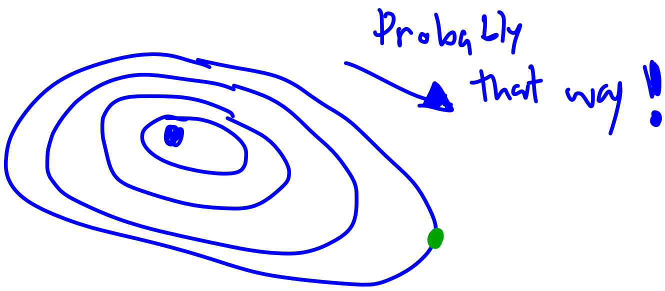
Plan: s → d → e → r → f → g

Cost: 11

Informed Search Strategies

Uninformed search | Informed search

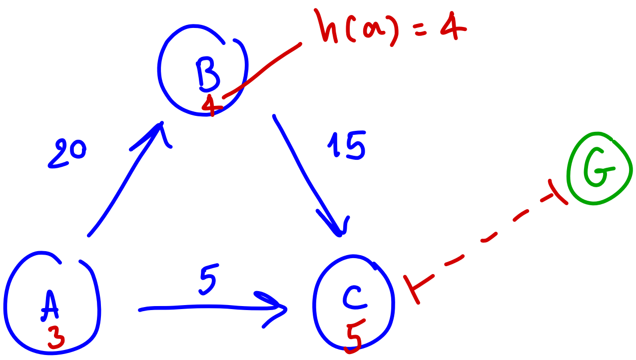




Heuristic Function: $h(s)$ – heuristic cost

$h^*(s)$ – estimated cost

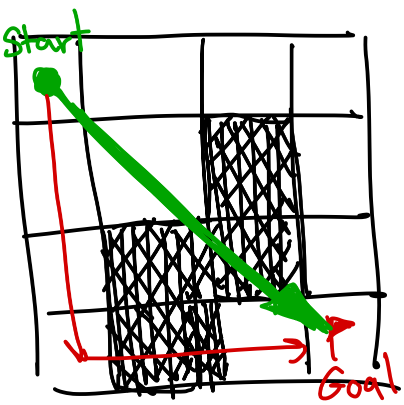
- Informed search
- estimate agent → goal
- not the best solution



$g(A \to B \to C) = 35$, $h(C) = 40$

$g(A \to C) = 5$, $h(C) = 40$

Example: Heuristic Function for Pathfinding



Euclidean distance

$$h(s) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan distance

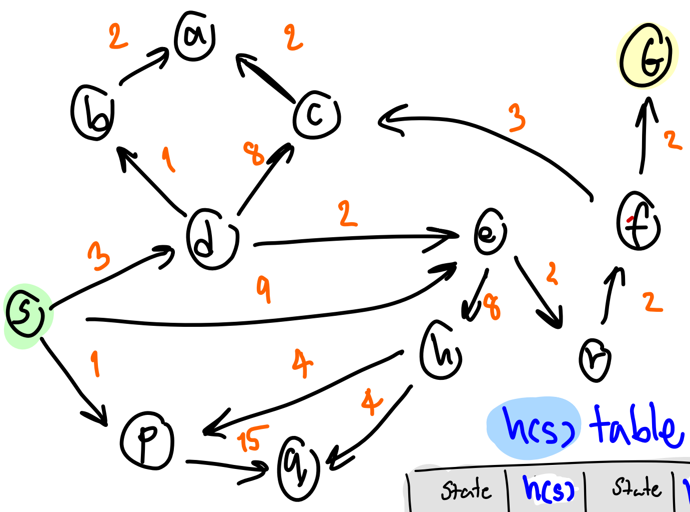
$$h(s) = |x_1 - x_2| + |y_1 - y_2|$$

ฮิวริสติก

Heuristic

Strategy 5: Greedy Best-First Search (GS)

Lowest h(s)



h(s) table

State	h(s)	State	h(s)
S	7	f	1.5
a	4	h	3
b	5.5	p	6
c	2	q	4.5
d	4.5	r	2.5
e	1.5	G	0

1. Completedness

~~Yes~~ No

* bad heuristic

2. Optimality

~~Yes~~ No

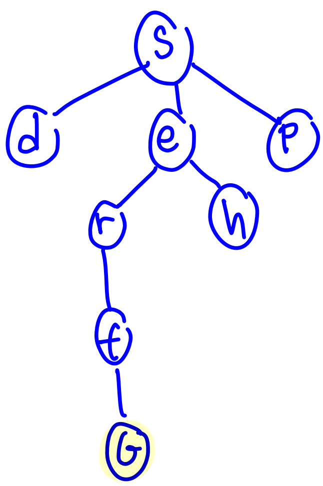
3. Time Complexity

$O(b^m)$

4. Space Complexity

$O(b^m)$

usually better if we have good heuristic

Frontier		Search Tree	Explored Set
Node	h(s)		State
S_1	7		S_1
d	4.5		e_2
e_2	1.5		r_3
p	6		f_4
r_3	2.5		G_5
h	3		
f_4	1.5		
G_5	0		

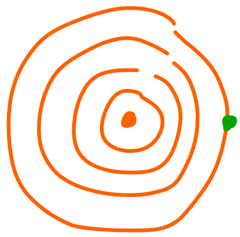
Plan: $S \to e \to r \to f \to G$

Cost: $7 + 1.5 + 2.5 + 1.5 + 0$

$= 12.5$

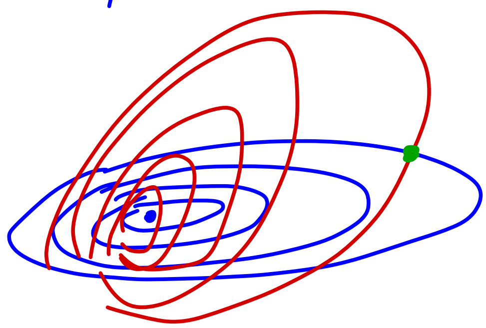
Path Cost Function + Heuristic Function : f(n)

Uniform Cost Search (UCS)



g(n)

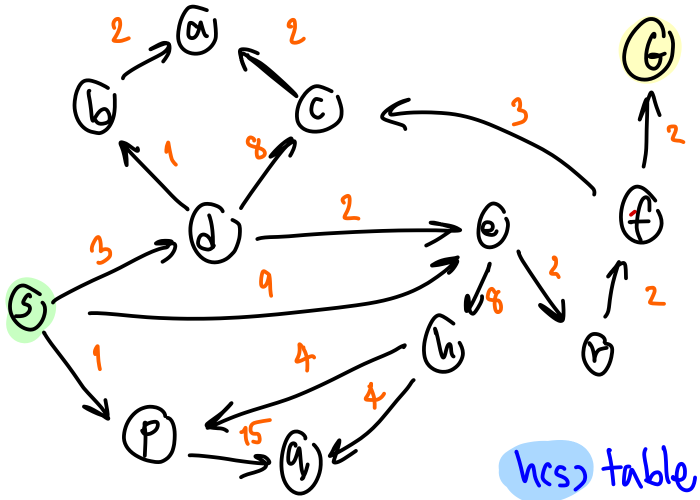
Greedy-best First Search (GS)



h(s)

$$f(n) = g(n) + h(s \text{ of } n)$$

Strategy 6 : A* Search Lowest f(n)



1. Completedness

   Yes

2. Optimality

   Yes

3. Time Complexity

   $O(b^{C^*/\varepsilon})$

4. Space Complexity

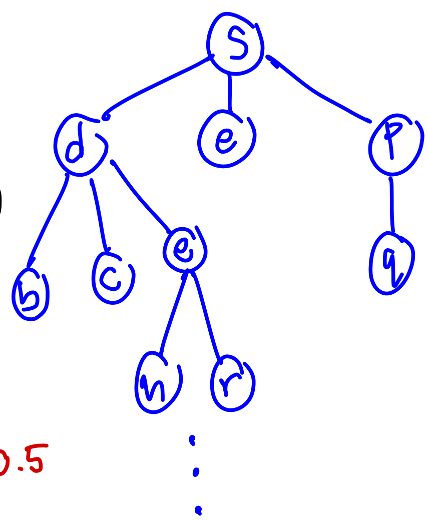
   $O(b^{C^*/\varepsilon})$

usually better with a good h(s)

h(s) table

State	h(s)	State	h(s)
S	7	f	1.5
a	4	h	3
b	5.5	p	6
c	2	q	4.5
d	4.5	r	2.5
e	1.5	G	0

$f(n) = g(n) + h(s)$

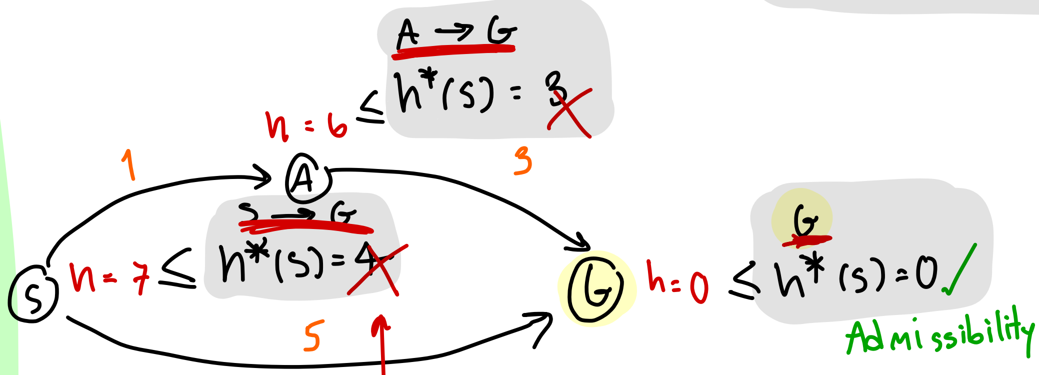
Frontier: Node	Frontier: f(n)	Search Tree	Explored Set: State
S 1	0+7		S 1
	g(n)+h(s)=f(n)		p 2
d 3	3+4.5 = 7.5		d 3
e	9+1.5 = 10.5		e 4
p 2	1+6 = 7		
q	(1+15)+4.5 = 20.5		
b	(3+1)+5.5 = 9.5		
c	(3+8)+2 = 13		
e 4	9+1.5 = 10.5		

Optimality of A* : Admissibility

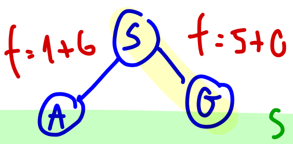
* h(s) is Admissibility if

$$0 \le h(s) \le h^*(s)$$

The minimum cost from s → G



must not over estimate the optimal cost



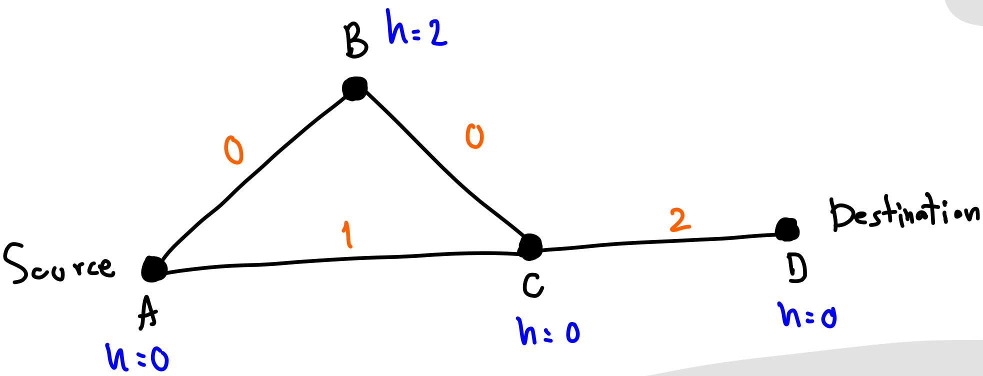
S → G = 5

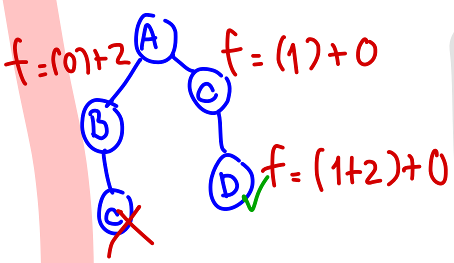
Optimality of A* : Consistency

* h(s) is Consistent

$$h(s) \le g(s \to s') + h(s')$$

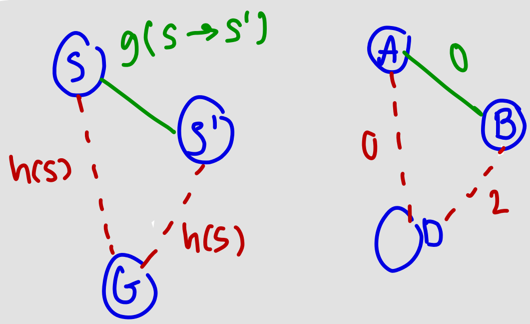
child of s





A → C → D = 3

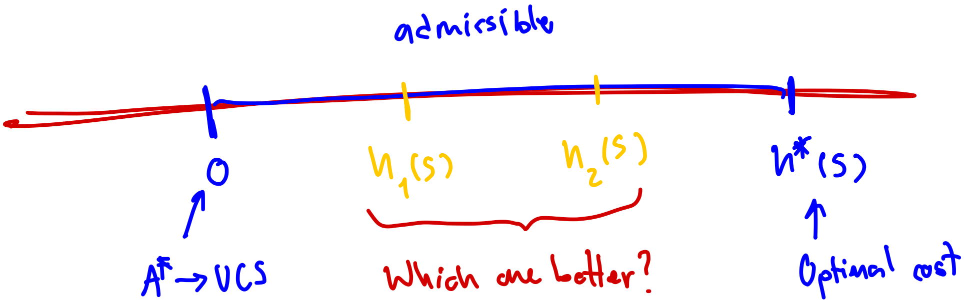
Example:



$h(A) \le g(A \to C) + h(C)$: $0 \le 1 + 0$ consistent ✓

$h(B) \le g(B \to C) + h(C)$: $2 \le 0 + 0$ not consistent ✗

$h_3(s) = h_1(s)/5$, admissible? — Question from last year.



Summary

Planning = Search Problem / Formulate → Graph Search / Strategy → A sequence of action

1. Initial State
2. Possible actions
3. Transition function
4. Goal test function
5. Cost function

1. DFS – Highest level
2. BFS – Lowest level
3. IDS
4. UCS – Lowest $g(n)$
5. GS – Lowest $h(s)$
6. A* – Lowest $f(n)$
   - Admissibility
   - Consistency