Three Stages of heuristic / informed:

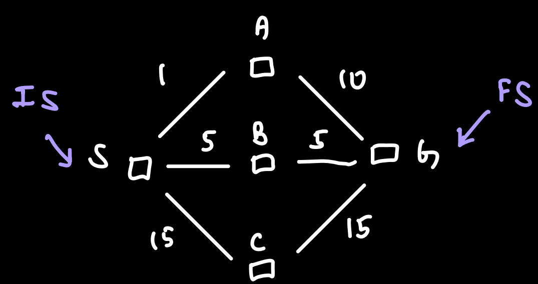
* Uniform - cost search

* Greedy (Best - first search)

UNIFORM - COST SEARCH (UCS):

Generalizes BFS

↳ allow arbitrary cost for action > 0.

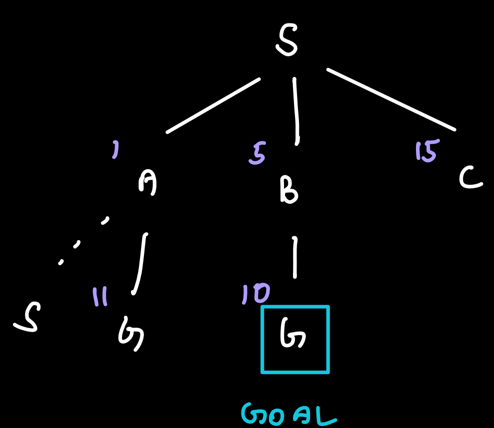


(state space)

Expand node

- check if goal

- if not, generate children.



$g(B) = 5$

$g(G) : 5+5=1$

S - B - G

(cost is 10)

If we had checked goal during generation, we would have ended at S-A-G with cost of 10.

* Completeness : yes
* Optimal : Yes, Lower cost before any higher cost.
* Time Complexity : BFS is subtype. So it is atleast as bad as BFS.

b^d

Suppose ε is the smallest cost of any action. If the optimal solution has cost c^*, then I may have to go down to depth $\lceil c^*/\varepsilon \rceil$

eg: $\varepsilon = 1/2$ $\quad c^* = 5$

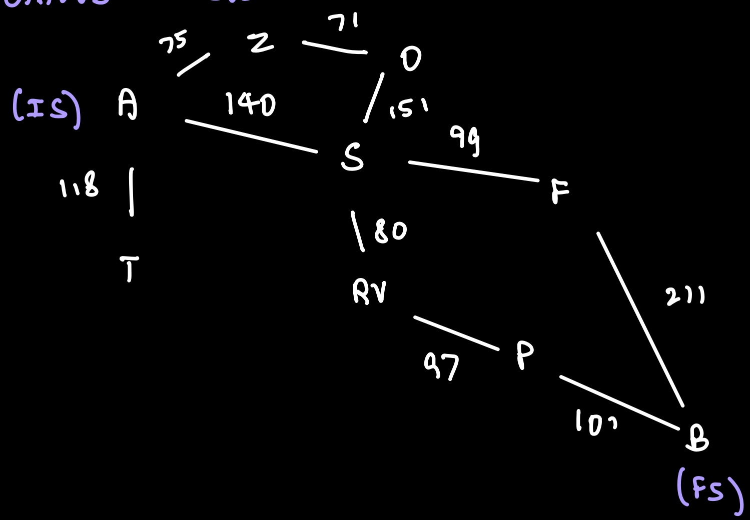
max. depth $= 10$

$b^{\lceil c^*/\varepsilon \rceil}$

* Space Complexity : $b^{\lceil c^*/\varepsilon \rceil}$

[Dijkstra's Algorithm]

GREED BEST - FIRST SEARCH :

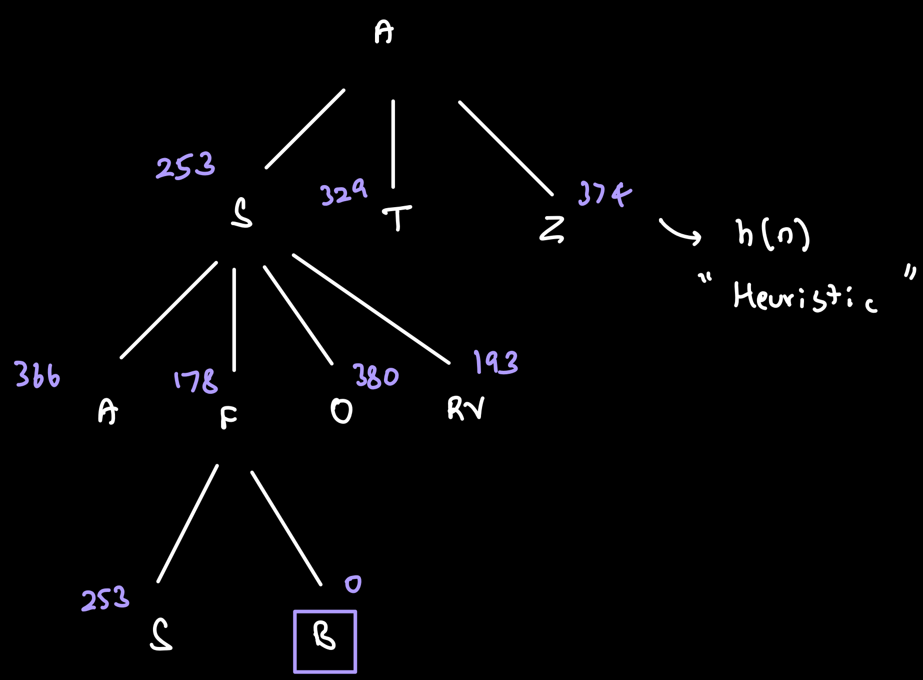


UCS will find

A - S - RV - P - B with cost = 418 //

GBFS needs a heuristic which is the "straight-line distance" from any point to FS.

	h(n)
A	366
S	253
RV	193
P	98
F	178
B	0
T	329
Z	374
O	380

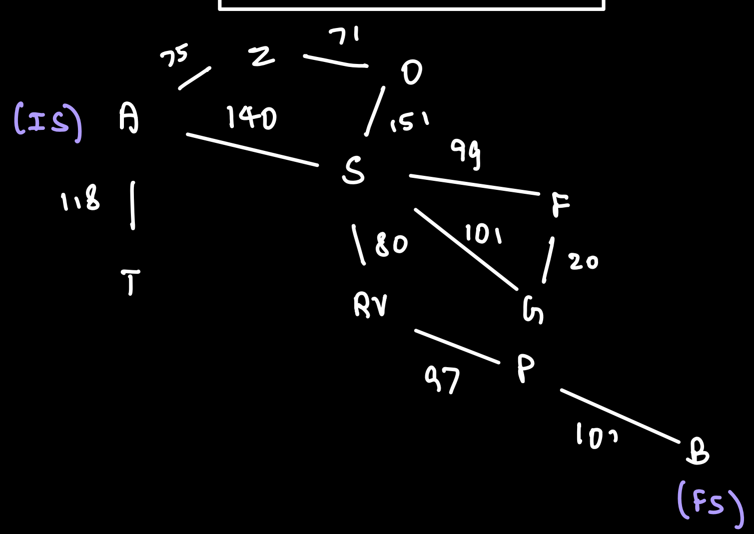


SOLUTION: A - S - F - B

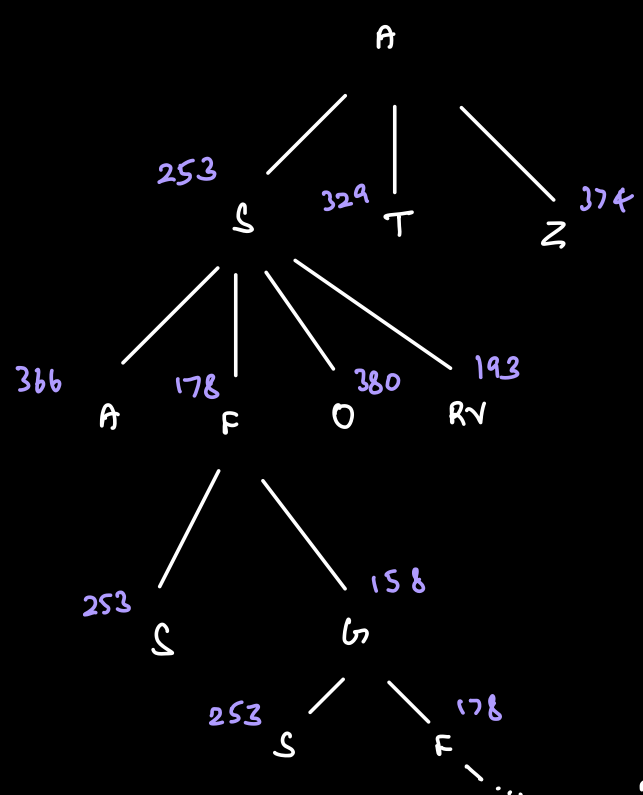
Cost 450 //

NOT OPTIMAL

NOT COMPLETE!



	h(n)
A	366
S	253
RV	193
P	98
F	178
B	0
T	329
Z	374
O	380
G	158



never completes?

UCS checks past → cost till now! [g(n)]

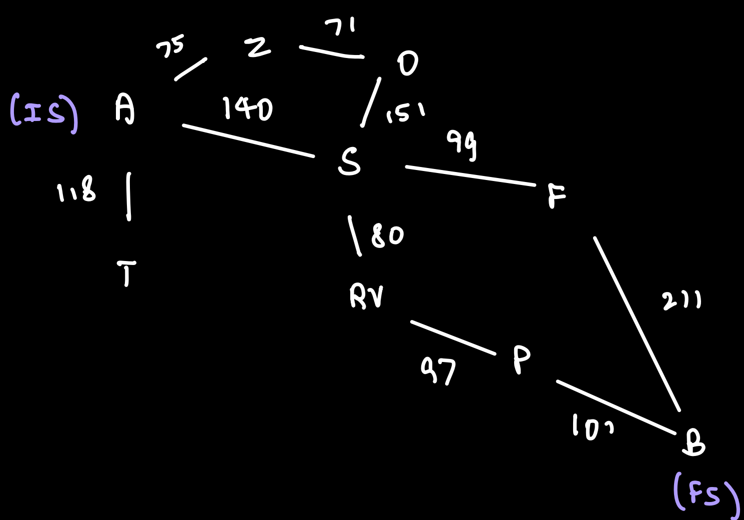
Greedy checks future → estimate of future cost. [h(n)]

A* SEARCH:

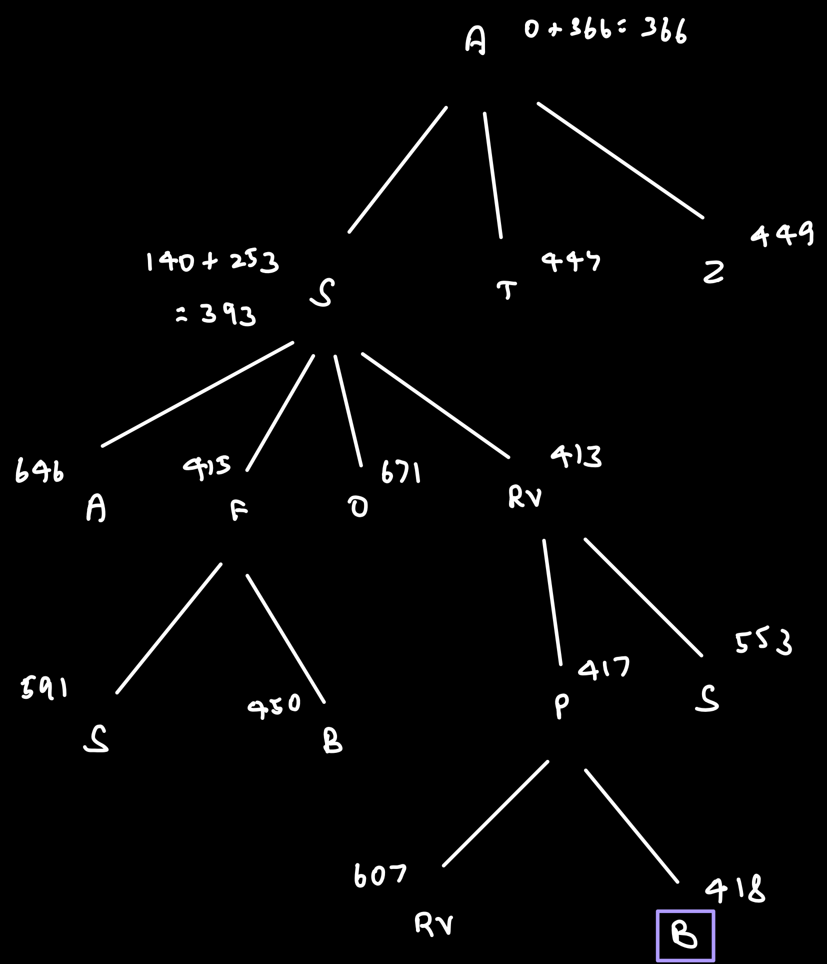
$$f(n) = g(n) + h(n)$$

↓

Evalution function.



	h(n)
A	366
S	253
RV	193
P	98
F	178
B	0
T	329
Z	374
O	380



Solution: A - S - RV - P - B

Evaluation Function:

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

$g(n)$: cost so far to reach n

$h(n)$: estimated cost to goal from n "heuristic"

$f(n)$: estimated total cost of a path that goes throug h.

Heuristic is "admissible"

$\longrightarrow h(n) \leq h^*(n)$ { least cost to go from n to goal }

Don't overestimate

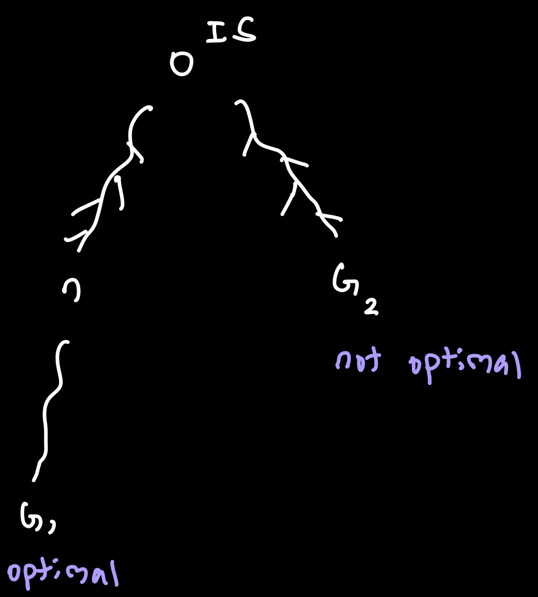
$\longrightarrow h(n) \geq 0$.

$h(G) = \phi$ for a goal state G.

A* with an admissible heuristic $(h(n))$ is OPTIMAL!

Higher the $h(n)$, faster the solution.

PROOF:



Assume 2 paths to FS (G), namely G_1, G_2.

G_1 is optimal. So, our algorithm is optimal, if for any intermediate node along the path IS - G_1, (including IS and G_1) n, n will be picked over G_2.

To PROVE: $f(n) < f(G_2)$

We know:

$$* \; f(G_2) = g(G_2) + \underbrace{h(G_2)}_{FS \,= 0}$$

$$= g(G_2)$$

$$* \; f(G_1) = g(G_1) + h(G_1)$$

$$= g(G_1)$$

G_1 is optimal

$$* \; f(G_1) < f(G_2)$$

$$\Rightarrow g(G_1) < g(G_2) \quad - 1$$

As n is along the path IS to G_1 and $h(n)$ is admissible

$$g(n_1) = g(n) + (\text{cost from } n \text{ to goal})$$

$$= g(n) + h^*(n)$$

$$\geq g(n) + h(n)$$

$$\geq f(n) \quad - ②$$

Combine ①, ②

$$f(n) \leq g(n_1) < g(n_2)$$

$$f(n) < g(n_2)$$

$$f(n) < f(n_2)$$

$$\downarrow$$

n is selected always!

HEURISTIC $h(n)$:

5	4	
6	1	8
7	3	2

1	2	3
8		4
7	6	5

$n \longrightarrow G$: goal state

Two heuristics (admissible):

h_1 : Number of misplaced numbers

All except '7', so $(1 \times 7) = 7$ //

h_2 : Manhattan distance of each number from original position to goal position

$2 + 3 + 3 + 2 + 4 + 2 + 0 + 2 = 18$

when $h_2(n) \geq h_1(n)$

and both are admissible

$h_2(n) \geq h_1(n) \geq h^*(n)$

h_2 dominates h_1 //

	Mean number of nodes expanded		
MINIMUM MOVES NEEDED (d)	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6
6	680	20	18
8	6384	39	25
12	364404	227	73
14	3473941	539	113
16	⋮	1301	211
⋮			
24		39135	164
	Uninforme	Informed	

↳ better heuristic.

What if h_1 and h_2 are not dominating each other?

$h_1(x) \geq h_2(x)$ for some x

$h_2(x) \geq h_1(x)$ for some other x

We choose

$$h(n) = \max(h_1(n), h_2(n))$$

$h(n) = 0$ is always admissible. This is nothing but Uniform Cost Search.