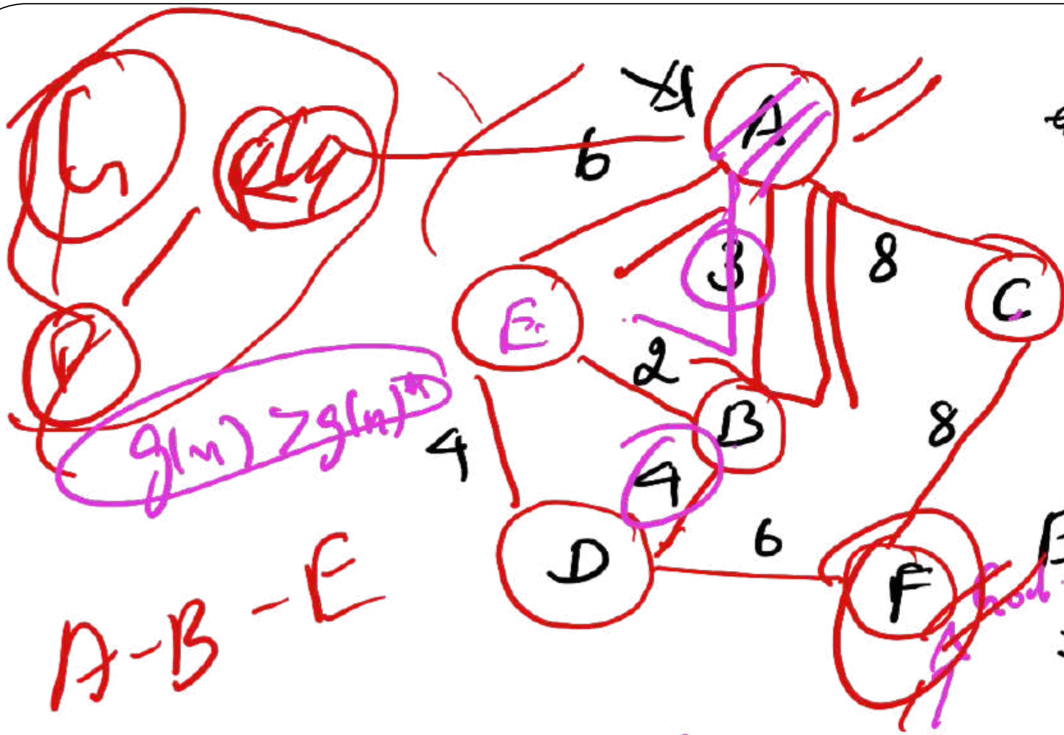
Artificial Intelligence



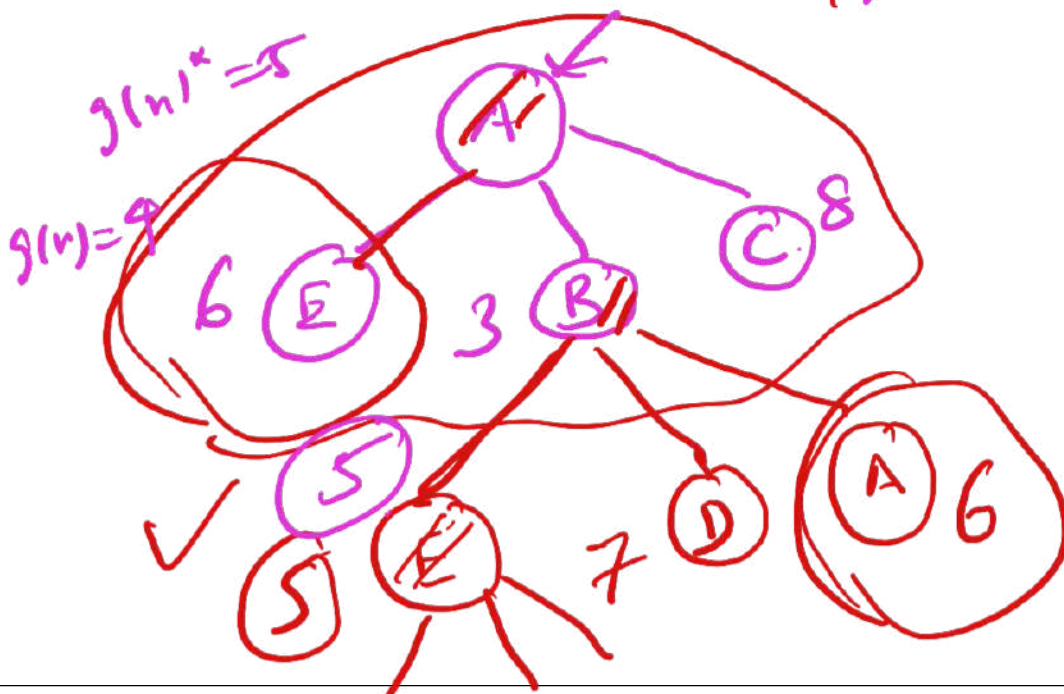
A-B-E

* Look for a Solution until

a) Complete solution is found

b) No other possible path with Smaller Cost.

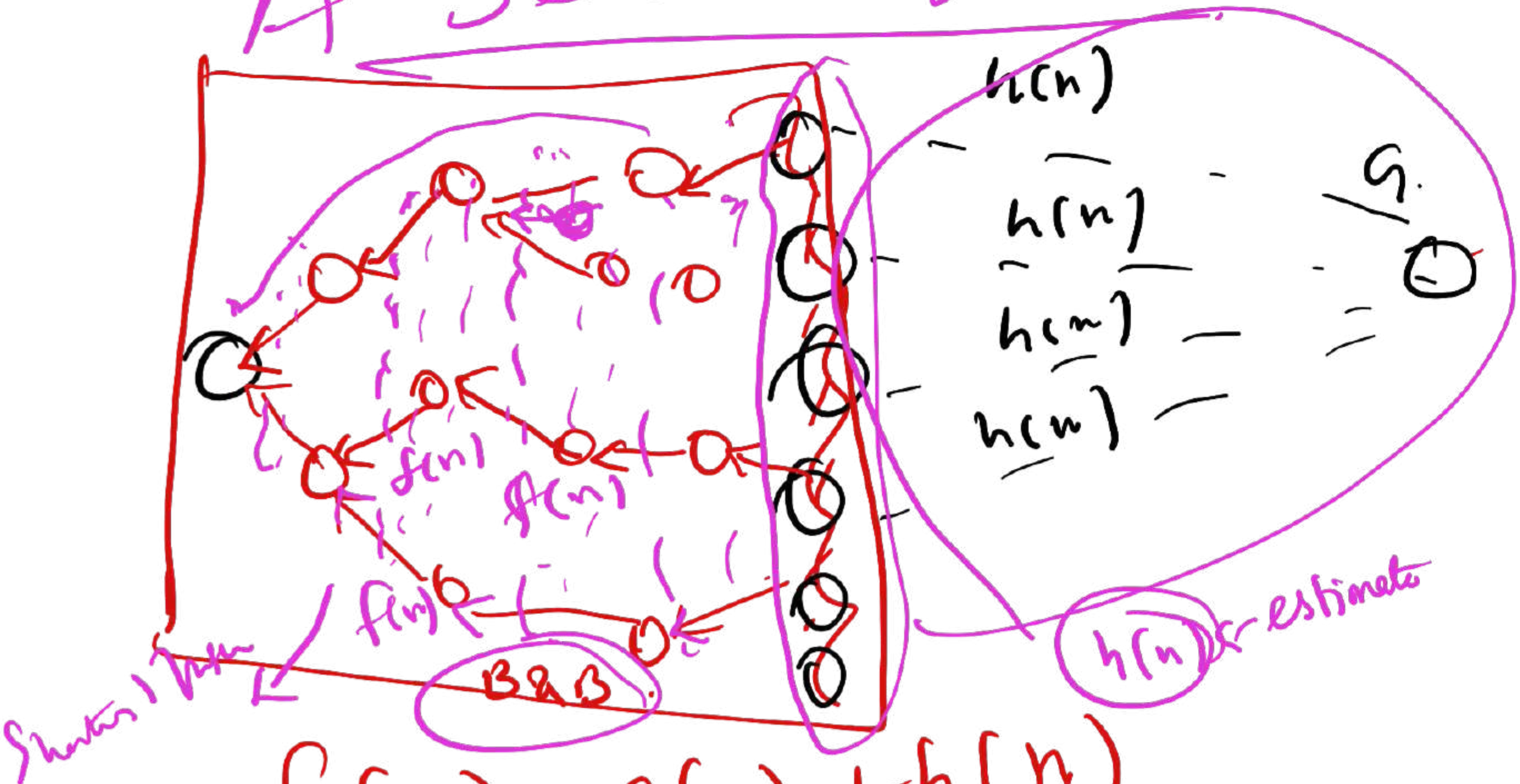
Estimated Cost (Full) Solution = Actual Cost of partial solution.



OPEN { A, E, D, C, A, B, D }

Closed { A, B, E }

A* Search Algorithm



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

$g(n)$ → Actual Cost of Path from S to n.

$h(n)$ → heuristic function from n to G.

Admissibility of A* :-

A* works with $f(n) = g(n) + h(n)$

but Djikstra's works with known costs $g(n)$ to node n.

Does A* always find a cheapest path?

Does it always find a path if there exists one

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

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

Optimal function

Admissable properties of A* :-

(1) $f(n)^*$ = optimal cost from source node to goal node via node n

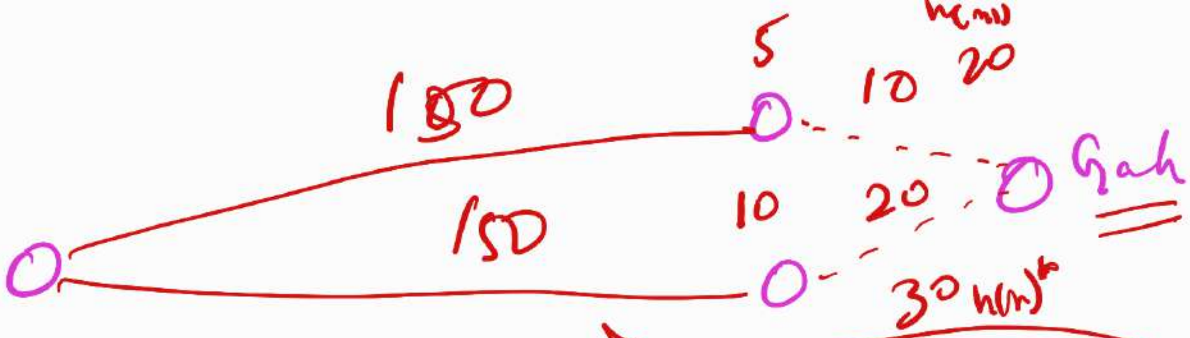
(2) $g(n)^*$ = optimal cost from source node to the node (n).

(3) $h(n)^*$ = optimal heuristic value from node (n) to goal state

To observe:-

(i) $g^*(n) \leq g(n)$; The algorithm will find optimal cost

(ii) $h(n) \leq h^*(n)$; heuristic function underestimate the distance to goal.



Weighted A*:- $f^*(n) = g^*(n) + h^*(n)$

$$f(n) = g(n) + w \times h(n)$$

$$w \to 0 \quad f(n) = g(n)$$

$$w \to \infty \to \text{non complete.}$$

A*:-

$$\underbrace{f(n)}_{\text{Estimated}} = \underbrace{g(n)}_{\text{Known Cost}} + \underbrace{h(n)}_{\text{Estimated}}.$$

$\underbrace{\text{Djikstra's (or) B\&B,}}_{\text{Actual Cost/Known Cost}}$ (or) $\underbrace{\text{Best First Search.}}_{h(n).}$

Optimal. $f^*(n) = g^*(n) + h^*(n).$

$\# f(n) = g(n) + \# h(n).$

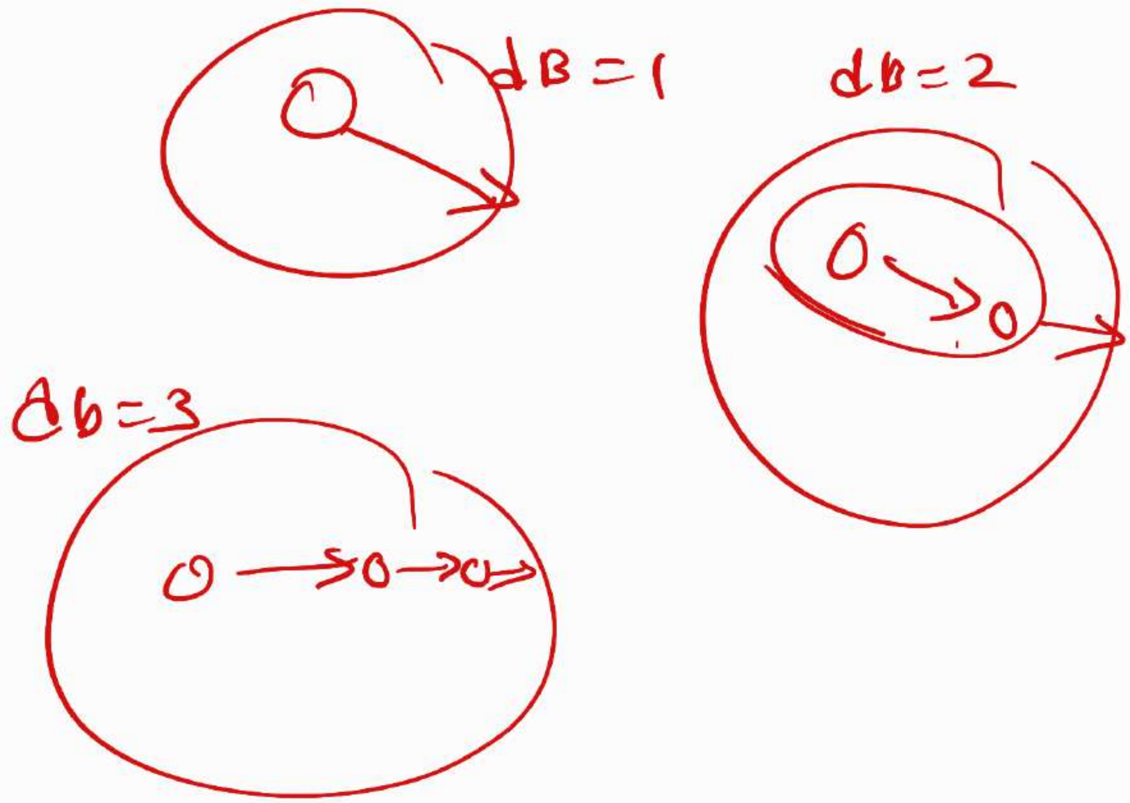
Iterative Deepening A*

* To Save on space by doing a Series of depth first Search of increasing depth.

* f-values → f(n).

* Initially, IDA*

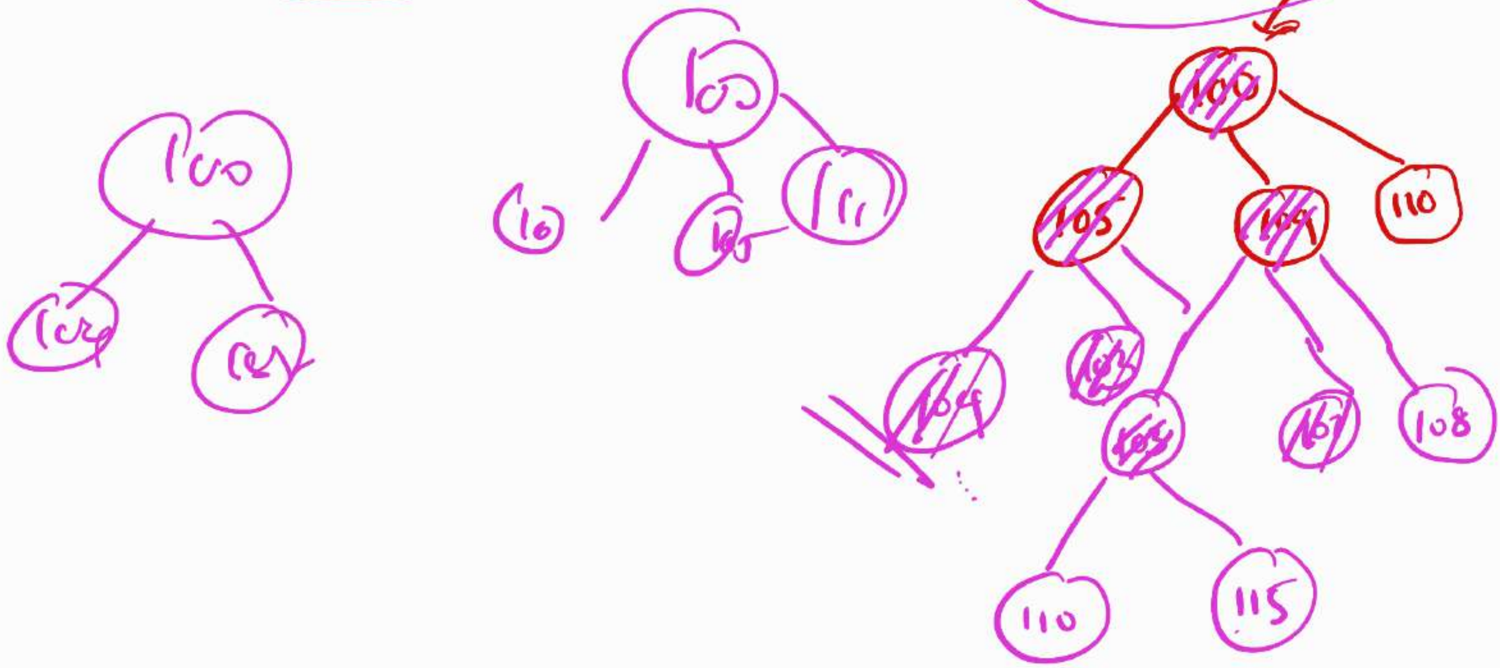
$$f(s) = h(s)$$ { Underestimate of optimal cost }



Pull & Push Approach

$f(n)$

$f(S) = 107.$



Iter Deepening A* (Start)

depthbound ← $h(S)$

While True.

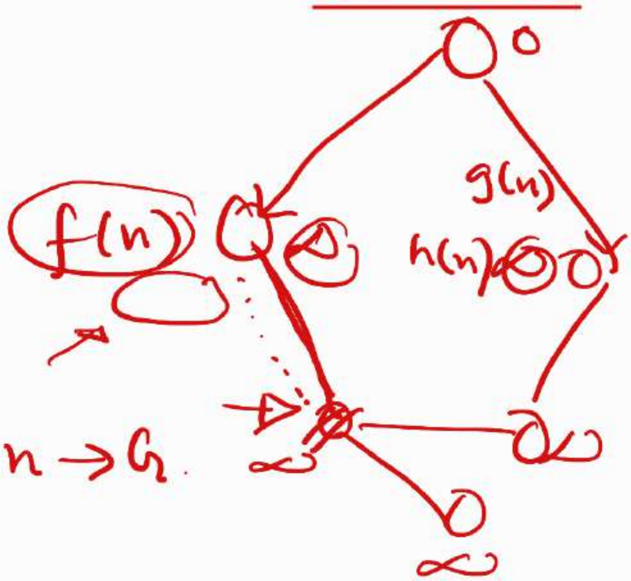
Do DepthBounded DFS (Start, Depthbound)

depthbound ← $f(S)$, where it is minimum of all unexplored nodes

$\rightarrow \quad h(n) = h^*(n)$

Estimate.

$f(n) = h(n) + 0$



f :- Case 1:-

Move gen( )

$\rightarrow f(n) =$ forward $n \rightarrow G$.

Case -2:-

Moving those states to open.

updating f(n) for all the open values

Case -3:-

Moving these states, f(n) of closed states

f(n) ← is also getting updated.

because of $\boxed{h(n)}$ ← Estimate value.

A*

⊥

optimal path

S

O

h(n)

O

f(n)

h(n) G

① $h(n) \geq h^*(n)$

Overestimating function

② $h(n) \leq h^*(n)$

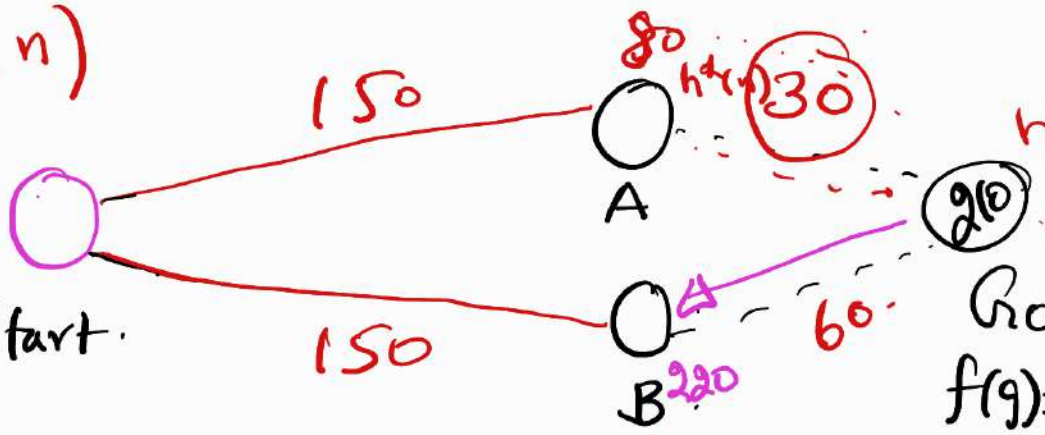
0.5

Underestimating function.

Case for Over-Estimation :-

$h_1(n) \geq h(n)$

$h_1(n) \geq h(n)$



$f(g) = g(g) + h(g) = 210 + 0 = 210$

$h_1(A) = \underline{80}$

$h_1(B) = \underline{70}$

$f_1(A) = h_1(A) + g_1(A) = 150 + 80 = 230$

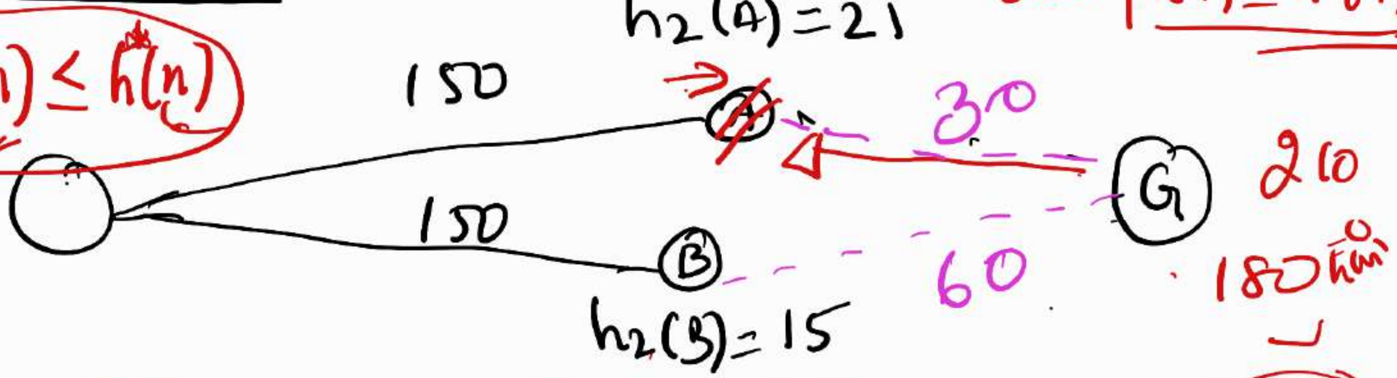
$f_1(B) = h_1(B) + g_1(B) = 150 + 70 = 220$

$f(g) = 210$

Underestimate :-

$h(n) \leq h^*(n)$ ✓

Ours $\boxed{h(n) \geq h^*(n)}$



$f(b) = h_2(B) + g_2(b) = 15 + 150 = 165$

$f(a) = h_2(A) + g_2(A) = 21 + 150 = 171$

$f(g) = h(g) + g(G) = 60 + 150 = 210$

Now 171 < 210, it Checks the path through A also → So it gives $= 150 + 30 = 180$ ← optimal

A* Admissiblity properties:-

1. State Space may be infinite; A* will work; branching factor should be finite.
2. $h(n) \leq h^*(n)$; $g(n) \geq g^*(n)$.
3. Choice of Choosing heuristic should be based on application.
4. The graph must have positive edges.