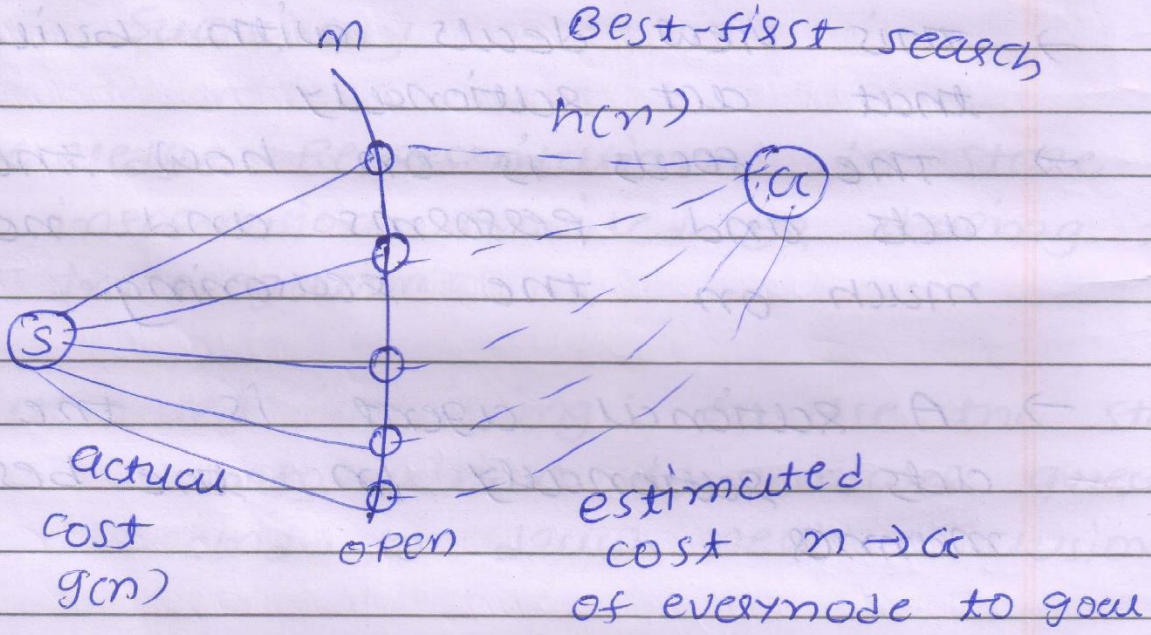
A* is admissible if

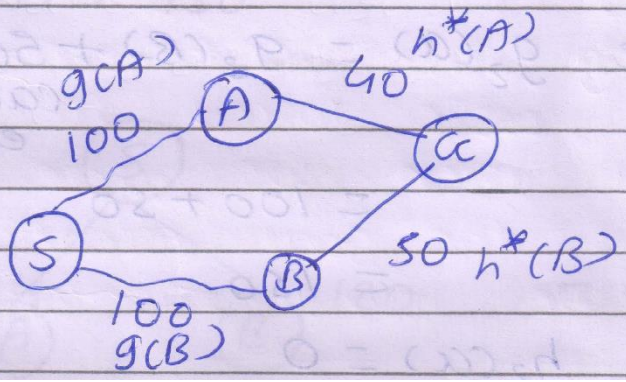
$h(n) \leq h^*(n)$ → underestimates

$h(n) \geq h^*(n)$ → overestimates

$h^*(n)$ → optimal (actual) cost of going from n to goal



→ Let say there are only two nodes in open A and B



Let say h_1 underestimates cost

h_2 overestimates cost

looking for function overestimate cost

$h_2(B) = 70$

$h_2(A) = 80$

$f_2(B) = 100 + 70 = 170$

$f_2(A) = 100 + 80 = 180$

→ A* maintains a priority queue of on f values

→ it will select node which has lowest f value

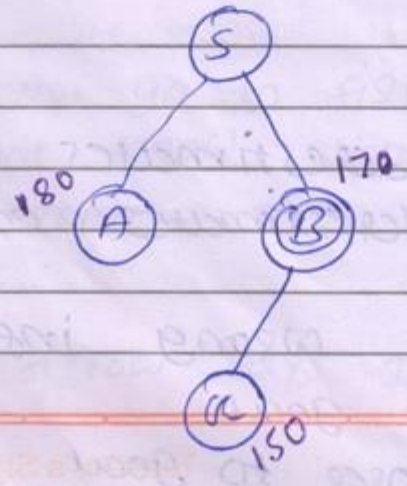
→ compute $g_2(a) = g_2(B) + 50$ (actual edge cost)

$= 100 + 50$

$= 150$

$h_2(a) = 0$

$f_2(a) = 0 + 150 = 150$



Teacher's Signature.....

→ NOW two nodes on open A and G

→ NOW $f_2(A) = 180$

$f_2(G) = 150$

→ So algo pick G and terminates

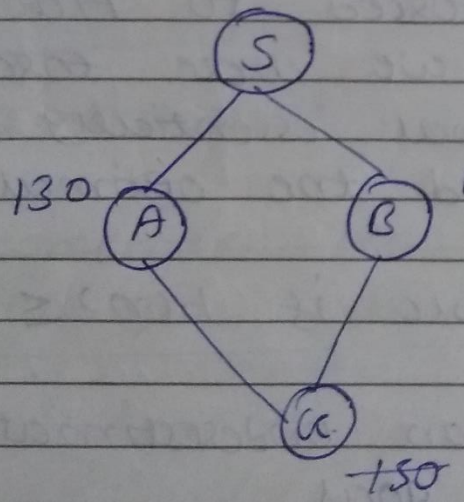
→ h_1 underestimates

$h_1(B) = 20$ // h_1 underestimates cost

$h_1(A) = 30$

$f_1(B) = 120$

$f_1(A) = 130$



- explore B first

- After picking B

$g_1(G) = 100 + 50$

$g_1(G) = 150$

$\therefore f_1(G) = 150$

$h_1(G) = 0$

→ NOW two node on open A and G

→ A* algo has to choose betⁿ G and A

→ A is lowes estimated cost 130 so, this algorithm will pick A and when pick A it will find cheapes cost to G

cheapes cost = 100 + 40 = 140

→ Using h_1 find optimal soln but using h_2 could not find the optimal soln

- Even though both of them thought that B was the better choice. Using h_2 we picked B and then we picked the goal

- using h_1 we picked B but then we were forced to pick A and then we were force to pick the goal essentially so, we found the optimal cost

∴ A^* is admissible if $h(n) \leq h^*(n)$

↓

heuristic function underestimates the cost of goal