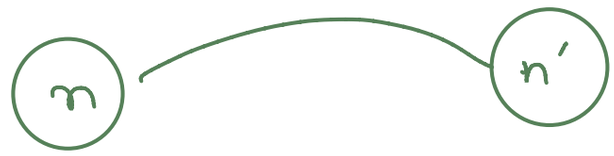
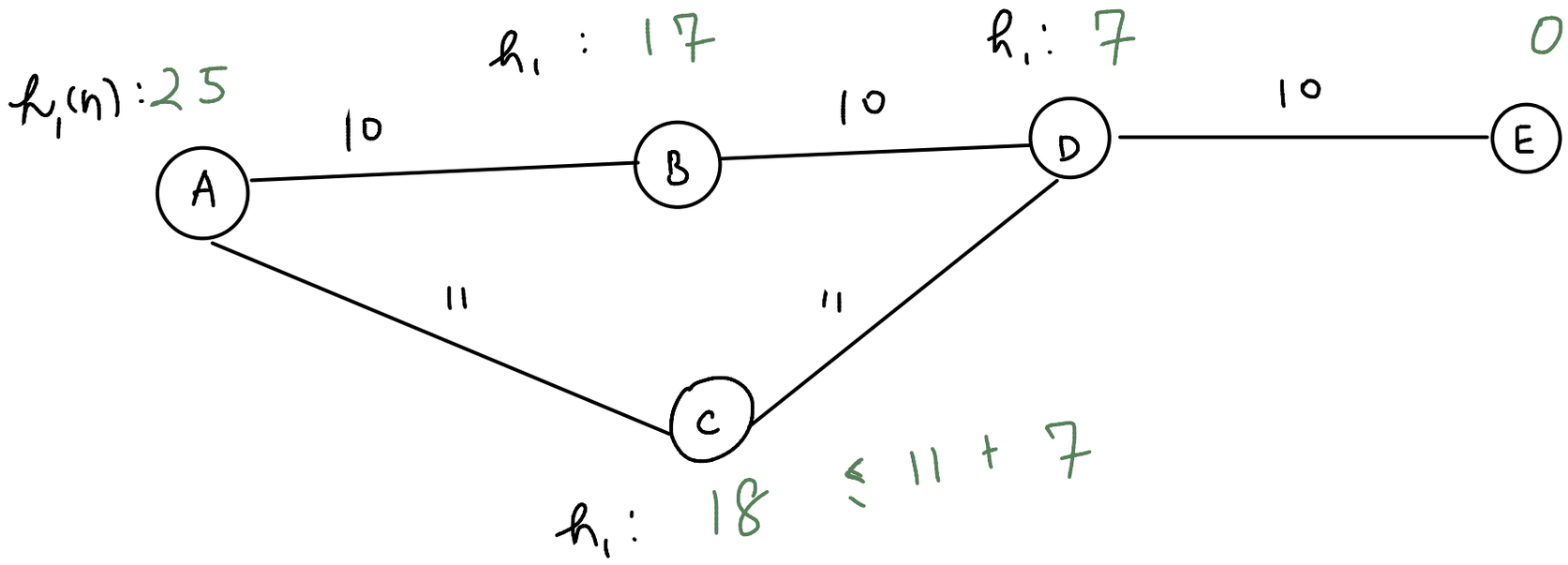
Admissible : $h(n) \leq$ total cost from node to Goal

Consistent : $h(n) \leq c(n, n') + h(n')$

step cost

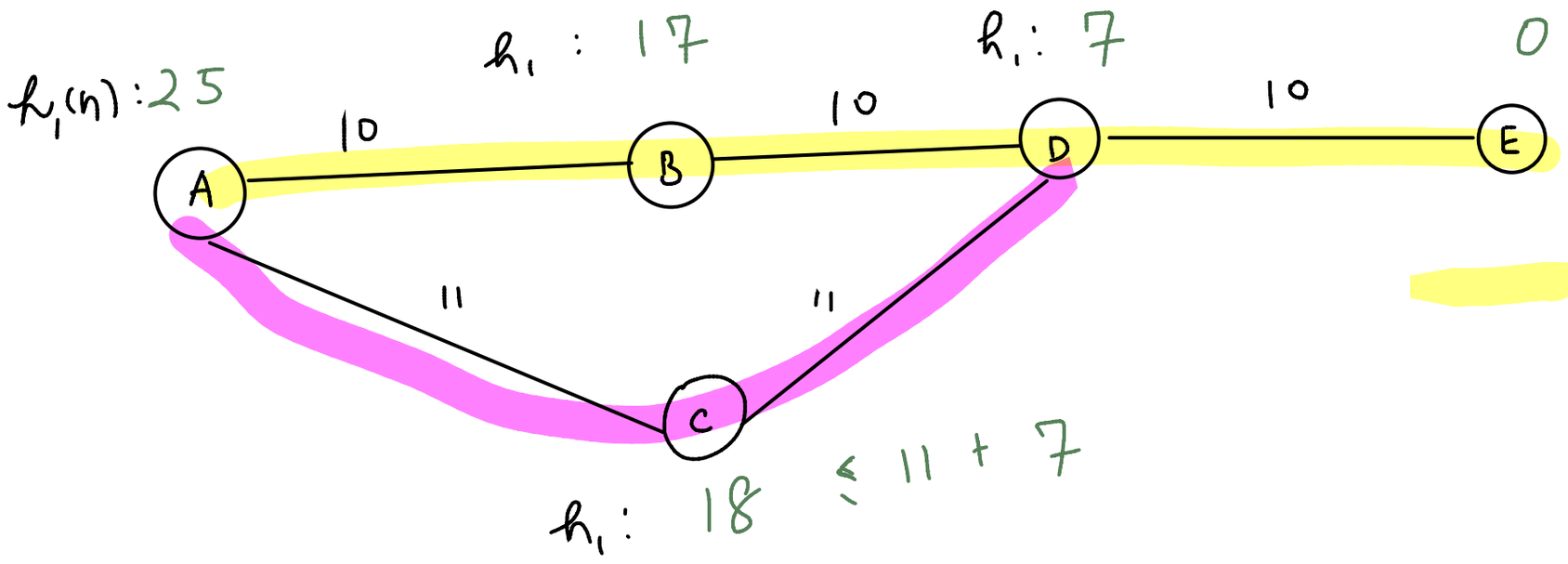


Fact : Consistent $\Rightarrow$ Admissible



h_1 is Consistent and hence admissible

Fact: If heuristic is Consistent Graph-Search will lead to optimal solution



Graph - Search

	Frontier	$g(n)$ Total cost till n	$h(n)$ Heuristic cost	$f(n) = g(n) + h(n)$	Top priority	Explored
				25		
Step 1	$n_1 = A$	0	25			
				27	n_1	A
Step 2	$n_2 = B$	10	17			
	$n_3 = C$	11	18	29		

Frontier	g(n) Total cost till n	h(n) Heuristic cost	f(n) = g(n) + h(n)	Top priority	Explored
Step 3					
$n_3 = C$	11	18	29	$n_2 = B$	A, B
$n_4 = D$	20	7	27		
Step 4					
$n_3 = C$	11	18	29	$n_4 = D$	A, B, D
$n_5 = E$	30	0	30		
× $n_6 = C$	31	18			

	Frontier	$g(n)$ Total cost till n	$h(n)$ Heuristic cost	$f(n) = g(n) + h(n)$	Top priority	Explored
Step 5	$n_5 = E$	30	0	30	$n_3 = C$	A, B, D, C

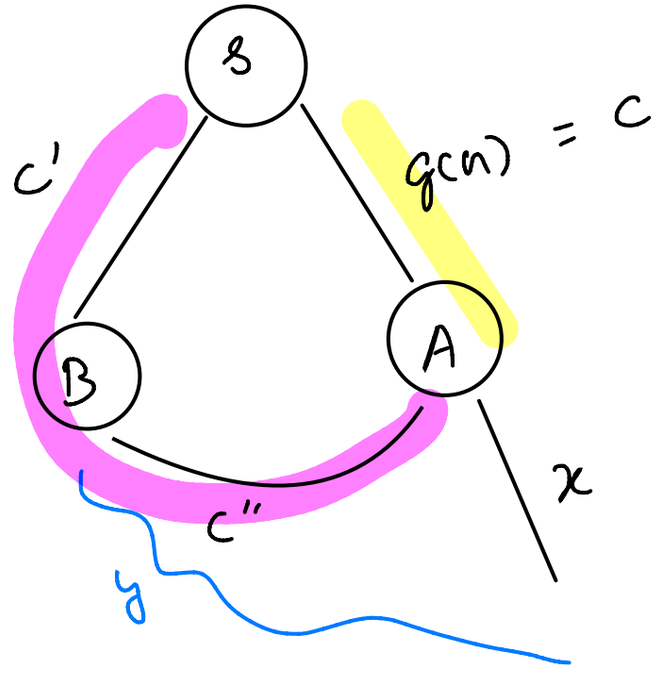
C does not add anything new specificially D is not taken up again

$n_5 = E$ is goal

blocks

Question: Can be better than

Ans : No. Consistency ensures that the very first time a node is explored, it is already via the optimal route



$y \leq c'' + x$

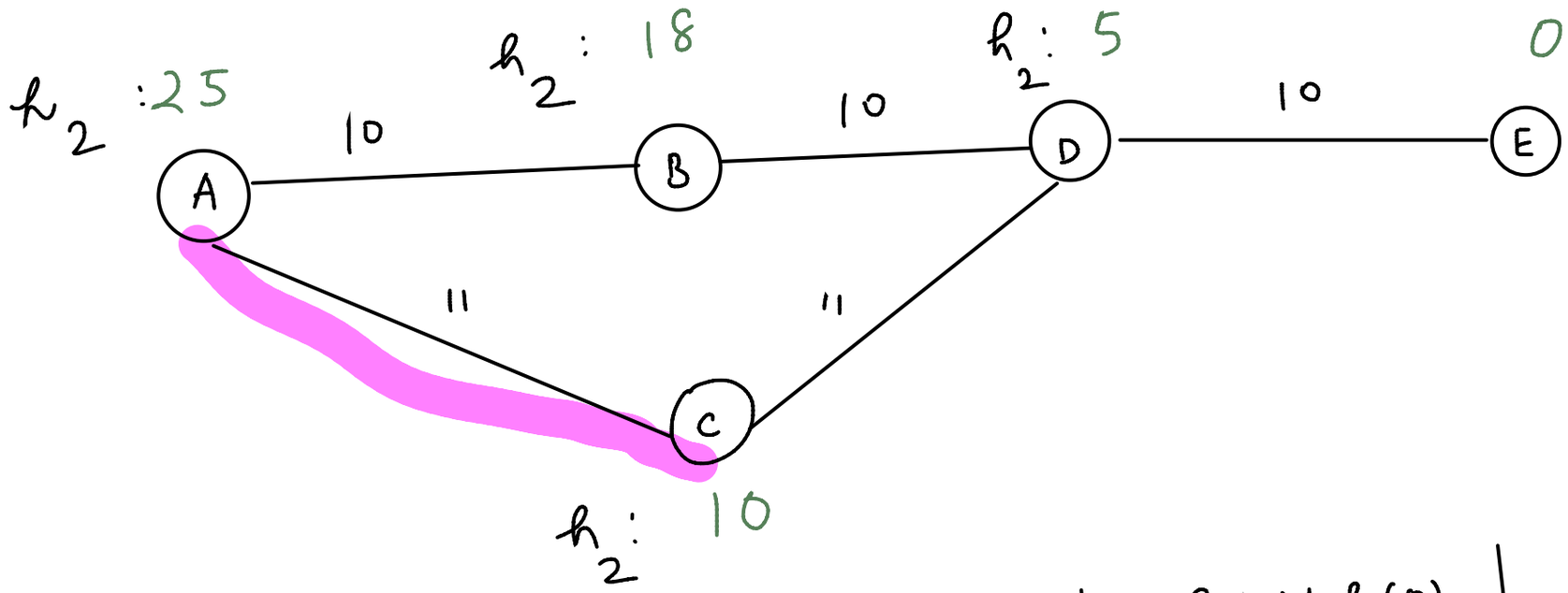
	n	g(n)	h(n)	f(n) = g(n) + h(n)
step k	A	C	x	C + x
taking up A for exploration	B	C'	y	C' + y ≤ C' + C'' + x

$$C + x \leq C' + y \leq C' + C'' + x$$

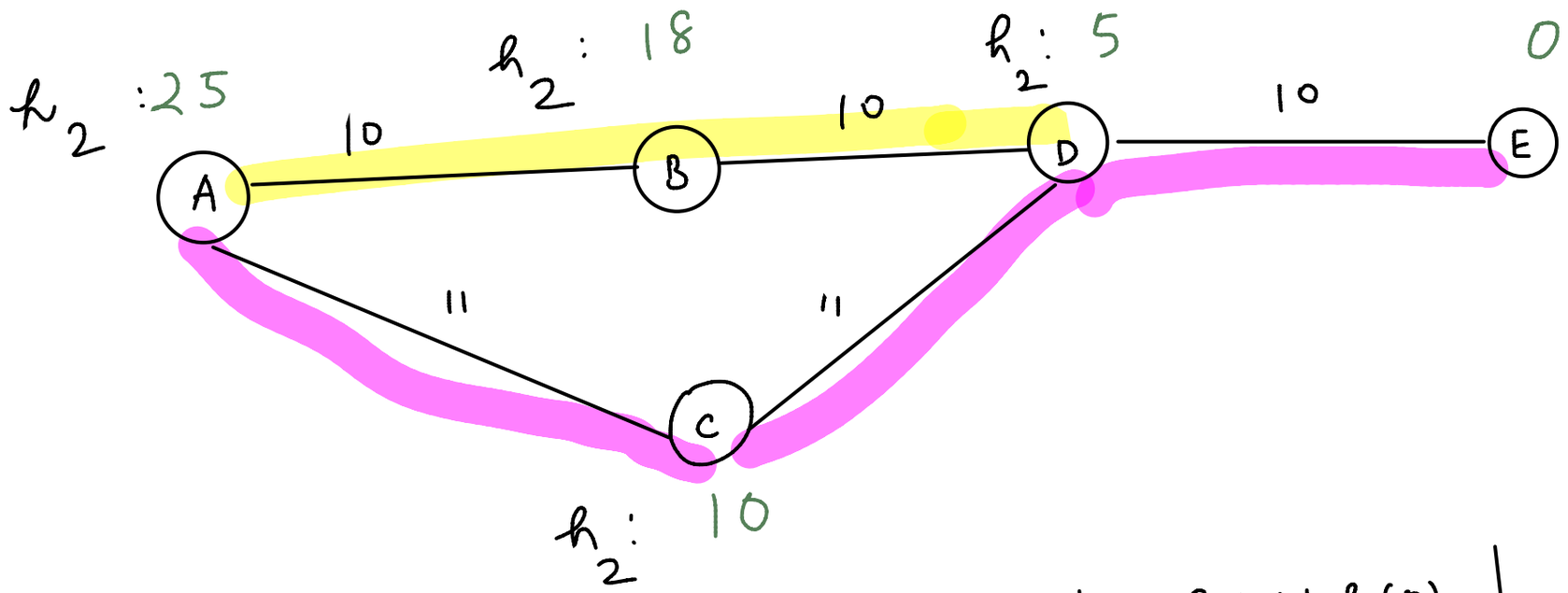
$$C \leq C' + C''$$

via B, A will be reached but it be blocked because it is already explored or part of frontier

First time A is taken for exploration

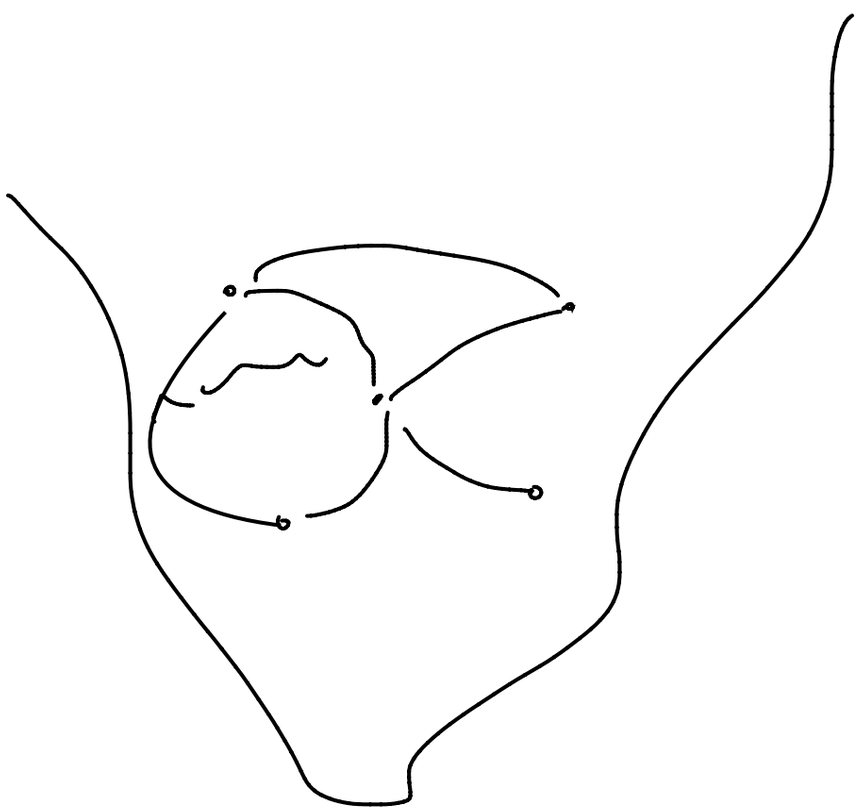


	n	g(n)	h(n)	g(n)+h(n)	Priority	Explored
Step 1	n_1 = A	25	0	25		
	n_2 = B	10	18	28	n_1	A
Step 2	n_3 = C	11	10	21		
	n_2 = B	10	18	28	n_3 = C	A, C
Step 3	n_4 = D	22	5	27		



	n	g(n)	h(n)	g(n)+h(n)	Priority	Explored
Step 3	n_2 = B	10	18	28	n_4 = D	A, B, C, D
	n_5 = E	32	0	32		
Step 4	B will not add any new node because blocks				n_2 = B	

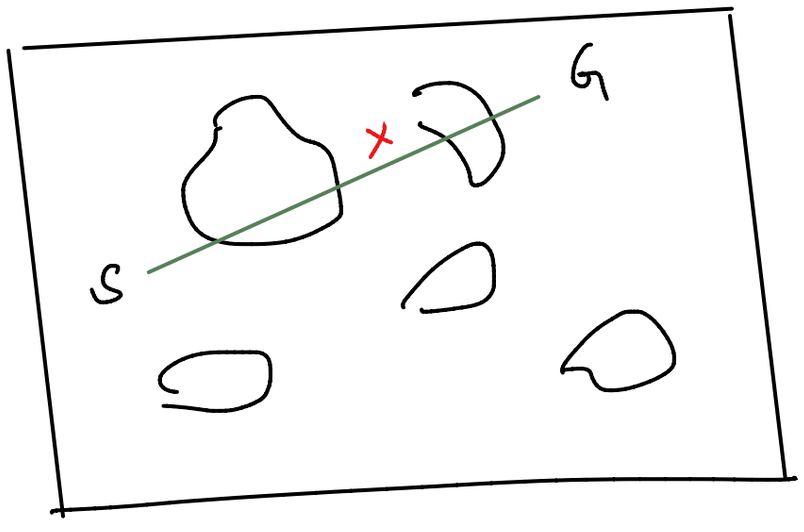
Where do heuristics come from



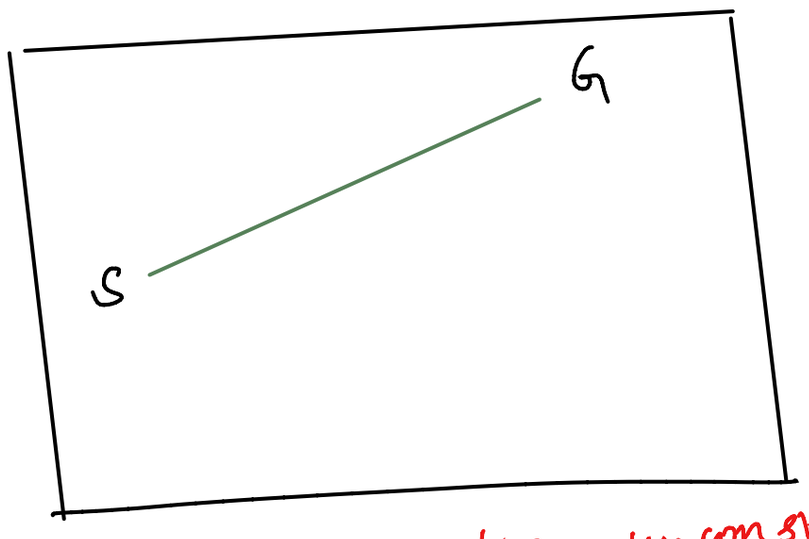
Train distance ≥

Actual road distance

≥ birds eye view distance



use it as heuristic in constrained problem



solve the unconstrained problem