Binary Tree Cameras Tryput - 1000 = [0.0, null, 0.0] Output - 1 Trituition Con 1 = return 0 if rode is not runifred Con 2 - return 1 if rode is monitred Con 3 - return 2 if hode has carruea Use Depth First Search (PFS) Bax con 1 - If mode = null return 1 int left = dFS (node, right, sun) int right = dFS (node, right, sun) - if (left == 2 right = 2) - //carreta kat limits null null return 1 - if (left == 0 right == 0) L - // node rund to be runifred sum+; return 2; - tf tagt other case So not runiford return 1; classmate		LEETCODE 968	
Tryst = 1000, null, 0, 0] Output = 1 Tritishon Cont 1 = return 0 if rode is not reasoning Cont 2 = return 1 if node is not reasoning Con 3 = return 2 if node has comera Use Depth First Scarch (PFS) Box cone 1 + If node == null return 1 int left = dFS (node, right, sum) lut right = dFS (node, right, sum) if (left == 2 right == 2) - comera is at luxuz node return 1 - if (left == 0 right == 0) L node neads to be reasoning return 2; return 2; Box cone 2 Only one node return 1; return 0; return 1;			
Trituition Care 1 = return 0 if rode is not remitted Care 2 = return 1 if node is nontread Care 3 = return 2 if hode has carriera Use Depth First Search (PFS) Box care 1 = If node == nell return 1 int left = dFS (node, right, sun) lut right = dFS (node, right, sun) if (left == 2 right == 2) - if (left == 0 right == 0) L. // node needs to be remitted return 2; return 2; return 2; return 0; Box care 2 Only one node return 1;			
Trituition Care 1 = return 0 if rode is not remitted Care 2 = return 1 if node is nontread Care 3 = return 2 if hode has carriera Use Depth First Search (PFS) Box care 1 = If node == nell return 1 int left = dFS (node, right, sun) lut right = dFS (node, right, sun) if (left == 2 right == 2) - if (left == 0 right == 0) L. // node needs to be remitted return 2; return 2; return 2; return 0; Box care 2 Only one node return 1;		Input - root = [0,0, null, 0,0]	
Cone 2 - return 0 if node is not monitored Cone 2 - return 1 if node is nontrad Cone 3 - return 2 if Norde has cornera Use Depth First Search (PFS) Bax cone 1 - If node == null return 1 int left = dFS (node, right, sum) lut right = dFS (node, right, sum)		Owput-> 1	× ~
Cone 2 - return 0 if node is not monitored Cone 2 - return 1 if node is nontrad Cone 3 - return 2 if Norde has cornera Use Depth First Search (PFS) Bax cone 1 - If node == null return 1 int left = dFS (node, right, sum) lut right = dFS (node, right, sum)	1		9
Con 1 = return 0 if rode is not remaind Case 2 - return 1 if node is monitored Con 3 - section 2 if node has comerce Use Depth First Search (PFS) Bax coup 1 - If node == null return 1 int left = dFS (node, right, sun) int right = dFS (node, right, sun) if (left == 2 right == 2) - // camera is at laware null null return 1 - if (left == 0 right == 0) - node needs to be remitted sum+; return 2; - teturn 2; Bax case 2 Only one node so not monitored return 0;		Intuition	
Case 2 - return 1 if node is nonthead Case 3 - return 2 if horde has camera Use Depth First Search (PFS) Bax case 1 - If mode == nelll return 1 int left = dFS (node, left sum) int reflet = 2 right = 2) - (left == 2 right == 2) - (left == 0 right == 0) L- node need to be numbered return 2; return 2; - if they other case Box case 2 Only one node so not munitive.) return 0; return 1;			Ba camera
Case 2 - return 1 if node is nonthead Case 3 - return 2 if horde has camera Use Depth First Search (PFS) Bax case 1 - If mode == nelll return 1 int left = dFS (node, left sum) int reflet = 2 right = 2) - (left == 2 right == 2) - (left == 0 right == 0) L- node need to be numbered return 2; return 2; - if they other case Box case 2 Only one node so not munitive.) return 0; return 1;		Cox1 = return 0 if node is not monitored	
Use Depth First Search (PFS) Bax case 1 - If node == null return 1 int left = dFS (node, left, sum) int right = dFS (node, right, sum) if (left == 2 right == 2) - if (left == 0 right == 0) L // node needs to be runified of sum ++; return 2; - if that other cases return 0; return 1;			
Depth First Search (PFS) Bax case 1 - If node == nell return 1 int left = dFS (node, left jum) int right = dFS (node, right, sum) int left == 2 right == 2) - if (left == 2 right == 2) - if (left == 0 right == 0) L- // node need to be nowlined Sum++; return 2; return 2; back case 2 Only one node so not munitived return 0; return 1;		Case 2 - return 1 if node is monitored (
Box case 1 - If mode == null section 1 int left = dFS (numbe, left sum) lut right = dFS (norde, right, sum) - if (left == 2 right == 2) - // camera is at lunix null return 1 - if (left == 0 right == 0) - // node needs to be numitived > Sum ++; return 2; - if they other cases So not movibered return 0; return 1;		Con 3 - veturn 2 if node has camer	
Box case 1 - If mode == null section 1 int left = dFS (numbe, left sum) lut right = dFS (norde, right, sum) - if (left == 2 right == 2) - // camera is at lunix null return 1 - if (left == 0 right == 0) - // node needs to be numitived > Sum ++; return 2; - if they other cases So not movibered return 0; return 1;		12 0 1 F 1 = 1 (0 = -2)	
int left = dFS (node, right, sum) int right = dFS (node, right, sum) if (left == 2 right == 2) //camera kat limits node node return 1 - if (left == 0 right == 0) L - // node needs to be nowthend Sum++; return 2; - it left other cases So not monitored return 0; return 1;		Use Depth Hist Search (DFS)	Region
int left = dFS (node, right, sum) int right = dFS (node, right, sum) if (left == 2 right == 2) //camera kat limits node node return 1 - if (left == 0 right == 0) L - // node needs to be nowthend Sum++; return 2; - it left other cases So not monitored return 0; return 1;		Bax care 1 - If node == nell	
intright = dFs (node, right, sun) if (left == 2 right == 2) - //camera ket Lower node return 1 - if (left == 0 right == 0) - // node need to be nowhed Sum++; return 2; - if that other cases so not monitored return 1;		seturn 1	5
intright = dFs (node, right, sun) if (left == 2 right == 2) - //camera ket Lower node return 1 - if (left == 0 right == 0) - // node need to be nowhed Sum++; return 2; - if that other cases so not monitored return 1;		int lett - dEs(male detisum) /	<u> </u>
if (left == 2 right == 2) // camera & est liverx need need need to be removed summer return 1 left == 0 right == 0) return 2; return 0; return 1;			
- //camera ket luwex norde norde null null return 1 - if (left == 0 11 right == 0) L. // norde needs to be numbered return 2; - teturn 2; Book care 2 Only one node so not munitared return 0; return 1;		IN regist = ars (nove; ny, sur)	9
- //camera ket luwex norde norde null null return 1 - if (left == 0 11 right == 0) L. // norde needs to be numbered return 2; - teturn 2; Book care 2 Only one node so not munitared return 0; return 1;		if (101+== 2 11 right===2)	
node return 1 - if (left == 0 nght == 0) L- // node needs to be nowthered return 2; back case 2 Only one node return 0; return 1;	\rightarrow	170=1	
return 1 - if (left == 0 right == 0) L. // node needs to be monthered sum++; return 2; - if (left == 0 right == 0) Box care 2 Only one node so not monitored return 0; return 1;		La // camera kat linera (null null	
- if (left == 0 right == 0) L // node needs to be monitored sum+; return 2; Boux care 2 Only one node return 0; return 0; return 1;		Pusite	
Le // node needs to be nowstreed sum++; return 2; Book care 2 Only one node return 0; return 1;			
return 2; Boux care 2 Only one node return 0; return 1;		-> cum	1++:
Teturn 0; Box care 2 Only one node so not monitored return 1;		L- // node needs to be monthed	
return 0; so not monitored return 1;		return 2;	
return 0;	→	11 Last Other cases Box care 2	
classmate			
		classmate	PAGE