Trappet   Trappe	LEETCODE   42	DATE
Intuition  For at any index, water topped = min(lips, Time)  - cures haight  Note that the prefix Sum & suffix Sum which gives height man (left) & haightnes (right) respectively  Step - for loop from i=0 to haight laught  Step - prefix = [0,1,1,2,2,2,2,3,3,3,3]  Step - suffix = [3 3 3 3 3 3 2 2 2 1]  Step - for loop again from i=0 to haight laught  int (respect = 0);	_	may be the
Intuition  For at any index, water topped = min(lips, Time)  - cures haight  Note that the prefix Sum & suffix Sum which gives height man (left) & haightnes (right) respectively  Step - for loop from i=0 to haight laught  Step - prefix = [0,1,1,2,2,2,2,3,3,3,3]  Step - suffix = [3 3 3 3 3 3 2 2 2 1]  Step - for loop again from i=0 to haight laught  int (respect = 0);		1 47
Intuition  For at any index, water topped = min(lips, Time)  - cures haight  Note that the prefix Sum & suffix Sum which gives height man (left) & haightnes (right) respectively  Step - for loop from i=0 to haight laught  Step - prefix = [0,1,1,2,2,2,2,3,3,3,3]  Step - suffix = [3 3 3 3 3 3 2 2 2 1]  Step - for loop again from i=0 to haight laught  int (respect = 0);	<b>T</b>	E and a parameter of the control of
Dutput = 6  O 1 0 2 1 0 1 3 2 1 2 1  Trataition  For at any index, water togged = prin(lyon, Imm)  - cures height  water property  prin(1.3) -0 = 1  prin(1.		
Dutput = 6  O 1 0 2 1 0 1 3 2 1 2 1  Trataition  For at any index, water togged = prin(lyon, Imm)  - cures height  water property  prin(1.3) -0 = 1  prin(1.	height = [0,1,0,2,1,0,1,3,2,	
Intuition  For at any index, water to appeal = min (lynn, thum)  - current haight  water to appeal = min (1, 2) -0 = 1  water to appeal = min (1, 3) -0 = 1  wa	1,2,1]	
Better solution  Step1 - for loop from i=0 to height.length  Step2 - preto = [0,1,1,2,2,2,2,3,3,3,3,3]  Step3 - suffice = [3 3 3 3 3 3 3 2 2 2 2 1]  Step2 - for loop again from i=0 to height.length  Step2 - preto = [0,1,1,2,2,2,2,3,3,3,3,3]	Output = 6	0 1 0 2 1 0 1 3 2 1 2 1
Better solution  Step1 - for loop from i=0 to height.length  Step2 - preto = [0,1,1,2,2,2,2,3,3,3,3,3]  Step3 - suffice = [3 3 3 3 3 3 3 2 2 2 2 1]  Step2 - for loop again from i=0 to height.length  Step2 - preto = [0,1,1,2,2,2,2,3,3,3,3,3]		
Better solution Using prefix Sum & suffix Sum workid gives height may (left) & heightner (right) respectively  Step! - for loop from i=0 to height length  Step2 - pretix = [0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix = [3 3 3 3 3 3 2 2 2 2 1]  Step4 - for loop again from i=0 to height laught int (rupped = 0;	Intuition	The proof of the same
Better solution Using prefix Sum & suffix Sum workid gives height may (left) & heightner (right) respectively  Step! - for loop from i=0 to height length  Step2 - pretix = [0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix = [3 3 3 3 3 3 2 2 2 2 1]  Step4 - for loop again from i=0 to height laught int (rupped = 0;	For at any index, wester	or toupped = min(1, x., )
Better solution Using prefix Sum & suffix Sum colvict gives  height max (left) & heightner (right) respectively  Step! - for loop from i=0 to height length,  Step2 - pretix = [0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix = [3 3 3 3 3 3 3 2 2 2 1]  Step4 . for loop again from i=0 to height length,  int (rupped = 0)		- curest haidst
Better solution Using prefix Sum & suffix Sum cohich gives height max (left) & heightners (right) respectively  Step! - for loop from i=0 to height length,  Step2 - pretix=[0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix=[3 3 3 3 3 3 3 2 2 2 2 1]  Step4 - for loop again from i=0 to height laught  int (rupped=0)	The state of the s	
Better solution Using prefix Sum & suffix Sum cohich gives height max (left) & heightners (right) respectively  Step! - for loop from i=0 to height length,  Step2 - pretix=[0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix=[3 3 3 3 3 3 3 2 2 2 2 1]  Step4 - for loop again from i=0 to height laught  int (rupped=0)	0-02	3) -1 = 0
Better solution Using prefix Sum & suffix Sum cohich gives height max (left) & heightners (right) respectively  Step! - for loop from i=0 to height length,  Step2 - pretix=[0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix=[3 3 3 3 3 3 3 2 2 2 2 1]  Step4 - for loop again from i=0 to height laught  int (rupped=0)	min (10	3)-0=1
Better solution Using prefix Sum & suffix Sum colvid gines height max (left) & heightness (right) respectively  Step! - for loop from i=0 to height.length  Step2 - pretix=[0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix=[3 3 3 3 3 3 3 2 2 2 1]  Step3 - for loop again from i=0 to height.laught  int (rapped = 0;	win (	ام را ميالاً دوا
Better solution Using prefix Sum & suffix Sum cohich gives height man (left) & heightner (right) respectively  Step! - for loop from i=0 to height.length,  Step2 - predix = [0,1,1,2,2,2,2,3,3,3,3]  Step3 - suffix = [3 3 3 3 3 3 2 2 2 1]  Step4 - for loop again from i=0 to height.laught  int (respect = 0;	<b>V</b>	30 01
Step! - for loop from i=0 to height.length Step? - pretto = [0,1,1,2,2,2,2,3,3,3,3] Step? - suffice = [3 3 3 3 3 3 3 2 2 2 1] Step? - for loop again from i=0 to height laught int (repped = 0;	TOward To The History	+ 4
Step! - for loop from i=0 to height.length Step? - pretto = [0,1,1,2,2,2,2,3,3,3,3] Step? - suffice = [3 3 3 3 3 3 3 2 2 2 1] Step? - for loop again from i=0 to height laught int (repped = 0;	Lynn = Landit III; Helman	
Step! - for loop from i=0 to height.length Step? - pretto = [0,1,1,2,2,2,2,3,3,3,3] Step? - suffice = [3 3 3 3 3 3 3 2 2 2 1] Step? - for loop again from i=0 to height laught int (repped = 0;	Better solution Using prefix	Sum & suffic Sum cohich aires
Step = predix = [0,1,1,2,2,2,2,3,3,3,3]  Step = suffix = [3 3 3 3 3 3 3 2 2 2 1]  Step = fr loop again from $i=0$ to height larger int (respect = 0;	height mor (left) & heightness ()	ight) respectively
Step = predix = [0,1,1,2,2,2,2,3,3,3,3]  Step = suffix = [3 3 3 3 3 3 3 2 2 2 1]  Step = fr loop again from $i=0$ to height larger int (respect = 0;	Step) - for loop from i=	o to height length
Step3 - suffix = [3 3 3 3 3 3 3 2 2 2 1]  Step3 - for loop again from i=0 to hosplar laught  int trapped = 0;	T - T	<u>-</u> *
Steps - for loop again from i=0 to hasgles laught	3/00/2 - preside 10,1,1,2	-, 2, 2, 2, 3, 3, 3, 3
int trapped = 0;	Step3 - suffix = [3 3 3 ]	3 3 3 3 2 2 2 1 ]
int trapped = 0;	Stepp - for Loop again of	nom i=0 to hosgla laugh
toupped += min(prefix[i], suffix[i]) - height[i]	Tocypies 1- 1741	(h. direci) , ballaci ) - restaur [1]
Steps) - Return toapped	Steps) - Return toapped	
classmate		PAGE

	DATE
	Optimed Approach
	Using Two pointers
	height = [0,1,0,2,1,0,1,3,2,1,2,1]  1 81ep1 - 1 = 0, x = height-length-1 < two portons
	Stop - Imox 20, mos 20 < to store max in bothy
	directions
	Step3 - trapped=0
0	The state of the s
1966	Steps - In all two positors (L \le 8) & onee  positore cross, exit so use while (L \le 8)
	Step 5  if (height[l] < height[s]) {// ensur sight has some log
	if (heightle) < heightles) {// some log
	if (height [1] > lms) ?
	lma = height [1] ; //store
Sing.	3 2 max with 2 left max
	else {
	trapped += Bonan - heightli;
	392 = Police La, 1, 2, 2 \$2, 2, 3, 3, 8
	L++;
***	Step 6 else {
	if ( height [ 87 \ Yman ) & //lef + how
	Lt.
9,33	z rmin= heightes worldy
	3 tappel == 1 max - heght [8]
	classmate 82-; Step7 Return toapped;