```
11. Container with most water:
Ile: height = [1,8,6,2,5,4,8,3,7]
Explanation: The above vertical lines are
represented by away [1,8,6,2,3,4,8,3,7]
In this case, the max area of water

(blue section) the container can contain is 49.
Brute force:
  * check every pair of lines

* calculate area = width * min (height[i], height[j])

* keep track of max area.
clairs solution of
public int max Azea Bautforce (int [] height) {
        Ent n = height. Length;
                                    (there is shall the
         int maxAzea = 0;
         for (int i = 0; i < n; i++) {0} - the
          fox( int 3 = its; 3 < n; 3 +4) {
              int area = (j-i) * Math. min (height[i],
                                 height [i]);
            maxAzea = Math. max (maxAzea, azea);
      return maxArea;
    Time \rightarrow 0(h^2) space \rightarrow 0(1)
```

optimized approach (Two pointers); use two pointers! left = 0, right = n-1 redeculate area between left and right: Move pointer with coraller height, because: " Increasing the smaller height may increase area. , moving the larger height won't help. class Solution {
public 3nd max Area (Ind [] height) { Ent left =0; To ANDIA , IN Hel Ent right = height. length -1; ent max Area = 0; while (left < right) { int area = (right - left) * Math. nin (height [left], height (zight]); marArea = Math. max (maxArea, area); af (height[left] < height [right]) { والاز يميد left ++; glerosarelgereans". "ditriclass right --; } else { return max Area; (2) and mond of 1. Longe come (n - co) the state fruits beneals for dials &

board bearing and water a see for

torock of soid

seloj nestes

Day Run: height = [1,8,6,2, 3,4,8,3,7] left = 0. right = 8 area = (8-0) × min (=1,7) =8 -) mar Area = 8 height [left] = 1 < height [right] = 7 > move left }, left =1, right =8 =7 x 7 = 49 -> maxAra = 49 area = (8-1) * min (8,7). height [left] = 8 > height [right] = 7 > move right -> 7 left = 1, right = 7 nea = (7-1) = min(8,3) = 6 +3 = 18-) max Area = 49

ic = perferon tolk

Result = 49.

3 (tryon a Afel) wider Time > o(n) space > o(1).