**Problem Name:** Container with most water

**Topics:**

**Companies:**

**Level:** Easy

**Language:** C++

**Problem Statement**:

**Input Format:**

**Output Format:**

**Constraints:**

**Examples:**

**Brute force Solution:**

**Explanation:**

* We have to find the max area.
* Here we are fixing one point and counting all the possible answers & storing the max one.

**Code:**

**Time Complexity**: O(n2)

**Space Complexity:** O(1)

**Optimized Solution:**

**Explanation:**

* Make two pointers one (l) pointing to the start of the height vector and the other (r) pointing to the end of the height vector.
* int l=0,r=height.size()-1;
* Iterate the height vector until l<r and at each iteration, find the total water between l and r by the below method:
  + For any given l and r, the **height** of the contained water depends upon the **minimum ( left height, right height)**.
  + If left height is **(lh = height[l])** and right height is **(rh = height[r])**, then the **height** of the contained water is **min(lh, rh)**.
  + Also the **width** of the contained water is equal to the difference between right and left index, that is **r-l**.
  + So the **total area of water** contained is **width \* height**, that is **(r - l) \* min(lh, rh)**.
* int lh=height[l];
* int rh=height[r];
* max\_area=max(max\_area,(r-l)\*min(lh,rh));
* The **maximum area of water** that can be contained in each iteration is the **answer**.
* max\_area=max(max\_area,(r-l)\*min(lh,rh));
* Also, after finding the area of water for a given l and r, increase *l* or decrease *r* according to whole height is smaller lh or rh.
* if(lh<rh) ++l

else --r

**Code:**

**Time Complexity**: O(n)

**Space Complexity:** O(1)