Problem

https://leetcode.com/problems/container-with-most-water/description/

You are given an integer array $height$ of length n. There are n vertical lines drawn such that the two endpoints of the ith line are $(i, 0)$ and $(i, height[i])$.

Find two lines that together with the x-axis form a container, such that the container contains the most water.

Return the maximum amount of water a container can store.

Notice that you may not slant the container.

Solution

1. Let $h = height$. Let $a = 0$, $b = n - 1$, $V = 0$
2. While $a < b$:

2.1. $V \leftarrow \max(V, (b - a)\min(h_a, h_b))$

2.2. If $h_a < h_b$ then $a \leftarrow a + 1$. Else if $h_a > h_b$, then $b \leftarrow b - 1$. Else if $h_a = h_b$, then you may choose either $a \leftarrow a + 1$ or $b \leftarrow b - 1$.

V is the answer.

It is easy to know that the time & space complexity are respectively $O(n)$ and $O(1)$.

Proof

Let $a < b$ and $a, b \in 0..n - 1$, $v = (b - a)\min(h_a, h_b) = (b - a)h_b$.

If $h_a > h_b$, then $\forall a' \neq a$ s.t. $a' \in 0..n - 1$, $v' = (b - a')\min(h_{a'}, h_b) \leq v$.

If $h_a < h_b$, then $\forall b' \neq b$ s.t. $b' \in 0..n - 1$, $v' = (b' - a)\min(h_a, h_{b'}) \leq v$.

If $h_a = h_b$, then $\forall a' \neq a$ s.t. $a' \in 0..n - 1$, $v' = (b - a')\min(h_{a'}, h_b) \leq v$ and $\forall b' \neq b$ s.t. $b' \in 0..n - 1$, $v' = (b' - a)\min(h_a, h_{b'}) \leq v$.

If $a = 0$ and $b = n - 1$ at first, then:

Step 1:

- If $h_a > h_b$, then we can rule out $n - 2$ different cases: $a' \in \{1,2, \dots, n - 2\}$ and $b = 1$, $v' \leq v$, and get a local max value v from another different case $a = 0$, $b = n - 1$.
- If $h_a < h_b$, then we can rule out $n - 2$ different cases: $a = 0$ and $b' \in \{n - 2, n - 3, \dots, 1\}$, $v' \leq v$, and get a local max value v belonging to another different case $a = 0, b = n - 1$.
- If $h_a = h_b$, then we can always rule out $n - 2$ different cases:
  - $a' \in \{1,2, \dots, n - 2\}$ and $b = 1$, $v' \leq v$, or
  - $a = 0$ and $b' \in \{n - 2, n - 3, \dots, 1\}$, $v' \leq v$.

  and gets a local max value v belonging to another different case.

Thus $n - 1$ different cases are considered.

Similarly, at step 2, we can always rule out $n - 3$ different cases and get a local max value belonging to another different case $a = 1$, $b = n - 1$ or $a = 0$, $b = n - 2$. Thus $n - 2$ different cases are considered.

…

At step k, we can always rule out $n - k - 1$ different cases. Thus $n - k$ different cases are considered.

…

Finally, at step $n - 1$, we can rule out 0 cases and get a local max value belonging to the unique case s.t. $b - a = 1$.

At this moment, in total, we have considered

$$(n - 1) + (n - 2) + \cdots + 3 + 2 + 1 = \frac{1}{2}\big(1 + (n - 1)\big)(n - 1) = \frac{1}{2}n(n - 1)$$

different cases.

From these n height values, if we arbitrarily choose 2 different position to form the required container, then there are

$$C_n^2 = \frac{n!}{m!\,(n - m)!}\Big|_{n=n,m=2} = \frac{n!}{2!\,(n - 2)!} = \frac{1}{2}n(n - 1)$$

Therefore, we have considered all possible cases. So, we can directly get the answer (i.e., the max volume) from the $n - 1$ local maximum values mentioned before. This algorithm is correct.