

LeetCode Question Proof Ideas

Benjamin Chen

July 18, 2023

11. Container With Most Water

Let $S(i, j)$ denote the maximum amount of water that the container can contain.

The naive brute force method checks all $S(i, j)$ for $1 \leq i \leq n, i \leq j \leq n$, which takes $O(n^2)$ time.

The pseudocode looks more or less like the following:

```
for (i = 1; i <= n; i++) {
  for (j = i; j <= n; j++) {
    maxWater = max(maxWater, S(i, j));
  }
}
```

The popular two pointer approach is correct because it only examines the largest possible case amongst cases with certain restriction.

Suppose that we are at the beginning of the program, we are given the array of numbers $arr[1..n]$.

Initially, we set $\text{maxWater} = 0$ as the starting point. It is okay to set it to 0 because it is the theoretical minimum of the amount of water.

The left index (pointer) is set to be $\text{left} = 1$, the right index (pointer) is set to be $\text{right} = n$.

Without loss of generality, we assume that $arr[\text{left}] \leq arr[\text{right}]$. If we only consider all the possible formations of containers which uses length $arr[1]$, then the maximum amount of water is determined by $S(1, n)$. (Other choices of the second length only makes the width less, but height is at most by $arr[1]$.)

Now, we can safely say that we have the maximum value amongst all $S(i, j)$ where $i = 1, i \leq j \leq n$. We assign the maximum of $S(1, n)$ and maxWater that is previously saved to maxWater . We have effectively finished the first iteration of the outer loop.

Now, we can treat the remaining part as a new question with the maxWater set to be $S(1, n)$. The same arguments apply.