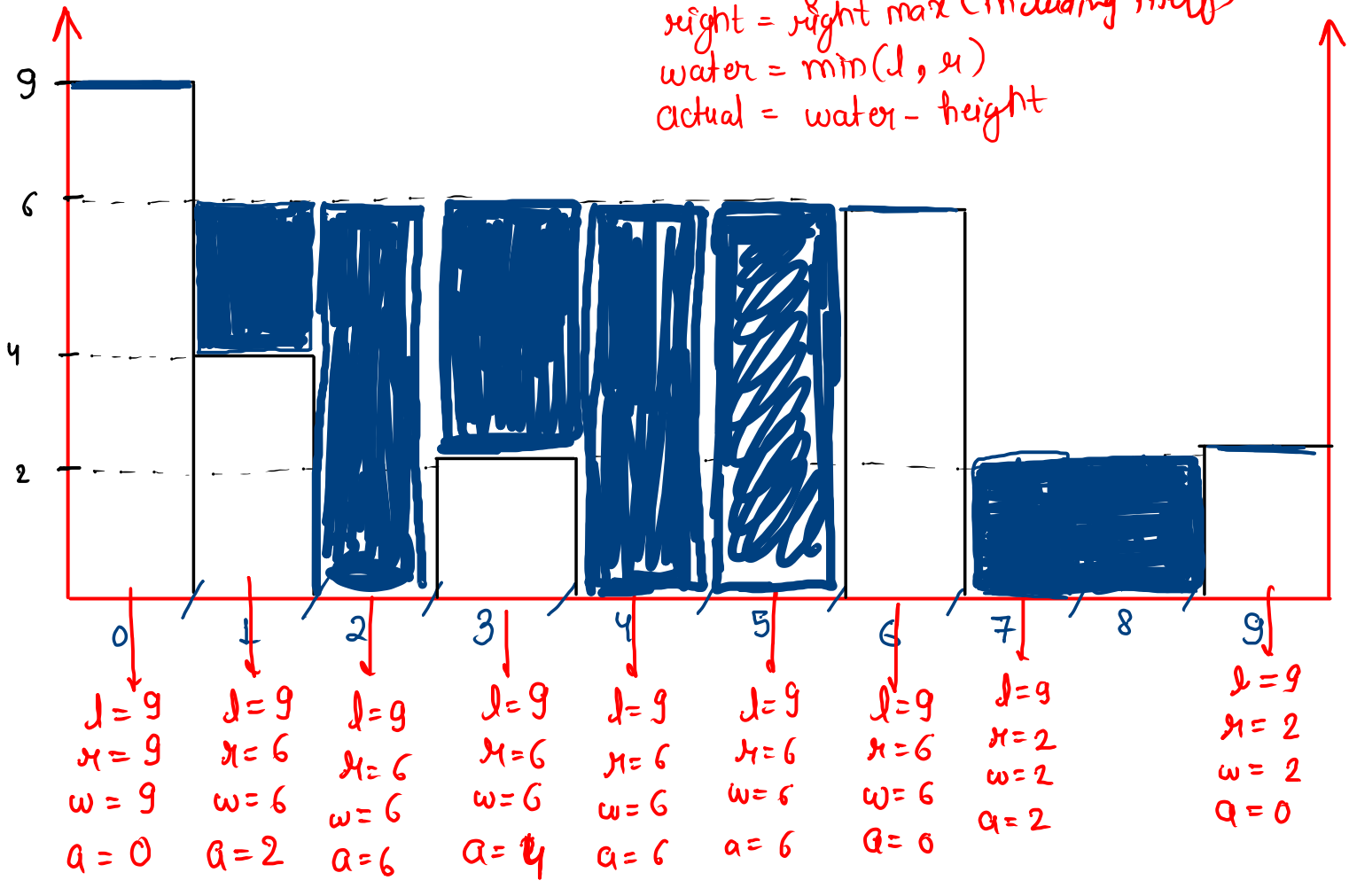
dry run Trapping rain water



psudo code

1) create ans = 0
2) traverse from start to end
   2.1) traverse from 0 to i^{th} index and find left max
   2.2) traverse from i^{th} index to $(n-1)^{th}$ index and find right max
   2.3) declare water = min(left, right)
   2.4) declare actualwater = water - curr. height (arr[i])
   2.5) ans += actualwater;
3) return ans

code

```java
public static int trappingRainWater(int[] arr, int n) {
    int ans = 0;
    for (int i = 0; i < n; i++) {

        int leftMax = Integer.MIN_VALUE;
        for (int j = 0; j <= i; j++) {
            if ( arr[j] > leftMax ) {
                leftMax = arr[j];
            }
        }

        int rightMax = Integer.MIN_VALUE;
        for (int j = i; j < n; j++) {
            if ( arr[j] > rightMax ) {
                rightMax = arr[j];
            }
        }

        int water = Math.min(leftMax, rightMax);
        int actualWater = water - arr[i];

        ans += actualWater;
    }
    return ans;
}
```

⇒ Time Complexity (time consumed by the program to get executed)

↳ T.C. will always be an approximation

Note:-

↳ T.C can only be calculated based on Number of operations performed

Ex 1 :-

```
main() {
    Syso("Hello");    // 1 operation
    Syso("World");    // 1 operation
}
```

Operation = 2 $\simeq$ **T.C = O(1)** → constant T.C

Note:- because no. of operations are not changing based on input that is why T.C is constant here

Ex2:-

```
main(){
    int n = scn.nextInt(); //5
    for(int i=0; i<n; i++){
        Syso("Hi");
    }
}
```

no. of operations

n = 5 ⟶ 5

n = 10 ⟶ 10

n = 100 ⟶ 100

⋮

n = n ⟶ n

T.C ∝ n

T.C = O(n)

linear T.C

Type of T.C

	Input	no. of operation
Constant →	n	1
linear →	n	n
Quadratic →	n	n^2
Cubic →	n	n^3
logarithmic →	n	log(n)

⇒ Rules of T.C

↳ all constant values with airthematic operator are always going to be ignored

Ex:-

no. of operations		T.C
$n+7$	$\simeq$	$O(n)$
$4*(n^3+3)$	$\simeq$	$O(n^3)$
$5*(n/2+7)$	$\simeq$	$O(n)$
$(n+1)^2$	$\simeq$	$O(n^2)$
(n^3+2n^2-n)	$\simeq$	$O(n^3)$

Rule :- In an expression, T.C will always pick variable with highest power

Ex :-

$$\left[4*\left((2*n)+(n/7)\right)- n*4\right] \quad T.C = O(n)$$