Rain Water Trapping

Data Structures & Algorithms

By: Shreerang Mhatre
Mukund Narsaria
Sarvesh Gurav
Aditya vishwakarma
Sakshi Vetotskar

Introduction:

DSA, or Data Structures and Algorithms, is a field of computer science that deals with the design and implementation of efficient data structures and algorithms. While DSA may not be directly related to the physical process of trapping rainwater, it can be used to design algorithms for simulating and optimizing rainwater trapping systems.

One example of using DSA for rainwater trapping is optimizing the placement of rainwater collection tanks in a given area. This can be modeled as a graph problem, where the nodes represent potential tank locations and the edges represent the distance between them. By using algorithms such as Dijkstra's algorithm or A* search, the optimal placement of tanks can be determined based on factors such as distance to rooftops, surface runoff areas, and stormwater drains.

Another example is using data structures such as arrays, linked lists, and stacks to efficiently store and process data from rain gauges and other sensors that measure the amount and intensity of rainfall. This data can be used to predict the timing and quantity of rainwater runoff, which in turn can be used to optimize the size and capacity of rainwater collection systems.

In summary, DSA can be used to design and optimize rainwater trapping systems by modeling them as graph problems and efficiently processing data from sensors and other sources.

</>

PROBLEM STATEMENT

Given an array of N non-negative integers arr[ ] representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

Explanation:

- To compute how much water an elevation map can trap after raining, we need to first understand the problem statement and the approach we can take to solve it.
- The problem states that we are given an array of N non-negative integers arr[], which represents an elevation map. We can visualize this elevation map as a set of bars of width 1, where the height of each bar represents the elevation at that point.
- After raining, some water may get trapped between these bars depending on the elevation of adjacent bars. We need to compute the total amount of water that can be trapped in this way.
- The approach to solve this problem is to iterate over each bar in the array, and for each bar, compute the maximum height of the bars to its left and right. The amount of water that can be trapped at this bar is then equal to the minimum of these maximum heights, minus the height of the current bar.
- Formally, let leftMax[i] and rightMax[i] be the maximum height of the bars to the left and right of the ith bar respectively. Then, the amount of water that can be trapped at the ith bar is given by:

  min(leftMax[i], rightMax[i]) - arr[i]
- We can then sum up this amount for each bar to get the total amount of water that can be trapped. To compute the leftMax and rightMax arrays efficiently, we can use two passes of the array. In the first pass, we compute leftMax from left to right, and in the second pass, we compute rightMax from right to left.
- The time complexity of this approach is O(N), as we only iterate over the array twice. The space complexity is also O(N), as we need to store the leftMax and rightMax arrays of size N

Code:

Code:

Code:

RESULTS:

Results here

CONCLUSION:

The benefits of collecting rainwater are numerous. It reduces the demand on the municipal water supply. It allows for storage of seasonal rains for use in off-peak times. Harvesting systems reduce erosion, property flooding, and contamination by reducing the majority of runoff from businesses and homes

References: