### Algorithm: Bucket Sort

To distribute treasure evenly among her crew, Princess Elara used the Bucket Sort algorithm to sort the values of treasure items.

#### Initialize Data Structures:

* Princess Elara used multiple buckets to categorize the treasure items.

#### Distribute and Sort:

* She distributed the treasure items into buckets and then sorted each bucket.

#### Implementation:

| **def** bucket\_sort(treasures: List[float]) -> List[float]:  n = len(treasures)  **if** n == 0:  **return** treasures  max\_value = max(treasures)  buckets = [[] **for** \_ **in** range(n)]  **for** treasure **in** treasures:  index = int(treasure / max\_value \* (n - 1))  buckets[index].append(treasure)  **for** i **in** range(n):  buckets[i].sort()  sorted\_treasures = []  **for** bucket **in** buckets:  sorted\_treasures.extend(bucket)  **return** sorted\_treasures  *# Example usage:*  treasures = [0.78, 0.17, 0.39, 0.26, 0.72, 0.94, 0.21, 0.12, 0.23, 0.68]  print(bucket\_sort(treasures)) *# Output: [0.12, 0.17, 0.21, 0.23, 0.26, 0.39, 0.68, 0.72, 0.78, 0.94]* |
| --- |

#### Explanation:

Initialize:

* buckets: Multiple buckets to categorize the treasure items.

Distribute and Sort:

### Princess Elara distributed the treasure items into buckets and sorted each bucket.