### Algorithm: Radix Sort

To manage her fleet's navigation charts, Princess Elara used the Radix Sort algorithm to sort coordinates of the sea monsters' locations.

#### Initialize Data Structures:

* Princess Elara used buckets (lists) for sorting based on each digit.

#### Sort by Each Digit:

* She sorted the coordinates digit by digit, starting from the least significant digit.

#### Implementation:

| **def** counting\_sort\_for\_radix(arr, exp):  n = len(arr)  output = [0] \* n  count = [0] \* 10  **for** i **in** range(n):  index = arr[i] // exp  count[index % 10] += 1  **for** i **in** range(1, 10):  count[i] += count[i - 1]  i = n - 1  **while** i >= 0:  index = arr[i] // exp  output[count[index % 10] - 1] = arr[i]  count[index % 10] -= 1  i -= 1  **for** i **in** range(len(arr)):  arr[i] = output[i]  **def** radix\_sort(locations: List[int]) -> List[int]:  max\_location = max(locations)  exp = 1  **while** max\_location // exp > 0:  counting\_sort\_for\_radix(locations, exp)  exp \*= 10  **return** locations  *# Example usage:*  locations = [170, 45, 75, 90, 802, 24, 2, 66]  print(radix\_sort(locations)) *# Output: [2, 24, 45, 66, 75, 90, 170, 802]* |
| --- |

#### Explanation:

Initialize:

* count: Buckets for sorting based on each digit.

Sort by Each Digit:

# Princess Elara sorted the coordinates digit by digit, starting from the least significant digit.