Counting Sort Cheatsheet

Overview

- Counting sort is an efficient sorting algorithm for small integers.
- It works by counting the number of occurrences of each element and using arithmetic to determine their position in the final sorted array.
- It has a linear time complexity of O(n + k), where k is the range of the non-negative key values.

Algorithm

```
def counting_sort(arr):
    # Find the maximum element in the array
    max_element = max(arr)

    # Create a count array of size max_element + 1 and initialize all elements to 0
    count = [0] * (max_element + 1)

    # Store the count of each element
    for element in arr:
        count[element] += 1

    # Modify the count array to store the actual position of each element in the
sorted array
    for i in range(1, max_element + 1):
        count[i] += count[i - 1]

    # Create the output array and fill it with the sorted elements
    output = [0] * len(arr)
    for element in arr:
        output[count[element] - 1] = element
        count[element] -= 1

    # Copy the sorted elements back into the original array
    for i in range(len(arr)):
        arr[i] = output[i]
```

Time Complexity

- Worst-case performance: O(n + k)
- Best-case performance: O(n + k)
- Average-case performance: O(n + k)

Resources

- Counting Sort Wikipedia
- GeeksforGeeks: Counting Sort
- Visualgo: Counting Sort