Bucket Sort

Introduction

Bucket Sort is a distribution-based sorting algorithm that works by partitioning the elements into different "buckets" or bins and then sorting each bucket individually. It is particularly useful when the input data is uniformly distributed over a range. Bucket Sort is named because it resembles the action of sorting balls into different buckets based on their values.

Logic

1. Create a fixed number of buckets, typically an array, each capable of holding a range of values.
2. Iterate through the input list and place each element in its respective bucket based on a specific mapping function.
3. Sort each individual bucket, either using another sorting algorithm or recursively applying Bucket Sort.
4. Concatenate the sorted buckets to obtain the final sorted list.

C++ implementation of Bucket Sort: Bucket Sort.
Python implementation of Bucket Sort: Bucket Sort.
Java implementation of Bucket Sort: Bucket Sort.
JavaScript implementation of Bucket Sort: Bucket Sort.

Complexity

- **Time Complexity**:
  - Best Case: O(n) when the elements are uniformly distributed across the buckets.
  - Average Case: O(n + n^2/k + k), where 'n' is the number of elements, and 'k' is the number of buckets.
  - Worst Case: O(n^2) when all elements are placed in a single bucket (e.g., due to a poor mapping function).
- **Space Complexity**:
  - O(n + k), where 'n' is the number of elements, and 'k' is the number of buckets.

Advantages

- Efficient for uniformly distributed data.
- Simple to understand and implement.
- Performs well when the range of input values is known in advance.

Considerations

- Requires an effective mapping function to distribute elements evenly into buckets.
- May require auxiliary sorting within buckets if there are many elements in a single bucket.
- Less efficient for non-uniformly distributed data.

Disadvantages/Limitations

- Inefficient for small datasets or datasets with a small range of values.
- Can be memory-intensive if the range of input values is large.

Edge Cases

- The efficiency of Bucket Sort depends on the choice of the number of buckets and the quality of the mapping function. Poor choices can lead to performance issues.

External Links

For more information on Bucket Sort, you can refer to the following articles: - Bucket Sort - Wikipedia - Sorting Algorithms: Bucket Sort - GeeksforGeeks - Bucket Sort - Visualgo

Bucket Sort is a useful sorting algorithm for specific scenarios, especially when you have a good understanding of the data's distribution and the range of values. However, it may not be the best choice for all situations, and other sorting algorithms like QuickSort or MergeSort might be more appropriate in many cases.