

نام و نام خانوادگی :

شایان سعیدیان

شماره دانشجویی :

401213015

تاریخ :

11/12/1402

عنوان گزارش :

تمرین 12 فصل 2

Alghoritm:

1. \*\*Initialization\*\*:

- Calculate the length of the input array `arr` and store it in `n`.

- Calculate the maximum possible value in the input array, which is determined by multiplying `k` (a constant multiplier) with `n` and adding 1 to it. This value represents the range of integers that could be encountered in the input array.

2. \*\*Count Array Initialization\*\*:

- Create an array named `count` to keep track of the occurrences of each integer in the input array. The size of this array is determined by the maximum possible value calculated in the previous step.

- Initially, set all elements of the `count` array to 0.

3. \*\*Marking Integers\*\*:

- Iterate through each element `num` in the input array `arr`.

- Use the values of the elements in `arr` as indices in the `count` array.

- Increment the corresponding count for each encountered integer by 1. This process effectively counts the occurrences of each integer in the input array.

4. \*\*Reconstruction of Sorted Array\*\*:

- Initialize an empty list to store the sorted elements, let's call it `sorted\_arr`.

- Iterate through the possible integers from 1 to the maximum possible value (inclusive).

- For each integer encountered, check if it has been marked in the `count` array.

- If the count of the current integer is greater than 0, it implies that the integer exists in the input array. In this case, add the integer to the `sorted\_arr`.

- After iterating through all possible integers, `sorted\_arr` will contain the sorted elements from the input array.

5. \*\*Return\*\*:

- Return the `sorted\_arr`, which now contains the sorted elements from the input array.

The counting sort algorithm sorts the input array by counting the occurrences of each integer and then reconstructing the sorted array based on those counts. It is efficient when the range of integers is relatively small compared to the size of the input array. The time complexity of the counting sort algorithm is O(n + int\_range), where n is the length of the input array and int\_range is the range of integers encountered.

=