274. H-Index

1. Given an array of integers `citations` where `citations[i]` is the number of citations a researcher received for their `i^th` paper, return *the researcher's h-index.*

   According to the definition of h-index on Wikipedia: The h-index is defined as the maximum value of `h` such that the given researcher has published at least `h` papers that have each been cited at least `h` times.

   Example 1:

   > **Input:** citations = [3,0,6,1,5]
   > **Output:** 3
   > **Explanation:** [3,0,6,1,5] means the researcher has 5 papers in total and each of them had received 3, 0, 6, 1, 5 citations respectively.
   > Since the researcher has 3 papers with at least 3 citations each and the remaining two with no more than 3 citations each, their h-index is 3.

   Example 2:

   > **Input:** citations = [1,3,1]
   > **Output:** 1
2. Took some time to understand what H-index is.
3. H-Index explained : https://www.youtube.com/watch?v=W6-PetKaCcA&t=258s

Solution using sort()

```
def hIndex(self, cit
    # [3,0,6,1,5]
    # [0,1,3,5,6]
    #  5 4 3 2 1
```

1.
2. Here at index 0..we have 5 papers that have more than citations..similarly at index 1 ..we have 4 papers with more than 1 citation…and at index 2 …we have 3 papers with more than 3 citations…

3. Python code

```python
class Solution:
    def hIndex(self, citations: List[int]) -> int:
        # [3,0,6,1,5]
        # [0,1,3,5,6]
        #  5 4 3 2 1
        N = len(citations)
        citations.sort()
        for i,v in enumerate(citations):
            if N-i <= v:
                return N-i
        return 0
```

Counting Sort()

1. We can solve this problem using counting sort in O(n)
2. Lets learn counting sort
3.