2- Counting Inversions and Soring

July 16, 2019

0.0.1 Counting Inversion and Sorting

Method: Divide and Conquer Input: an array of unsorted numbers
Output:

```
- # of pairs of array entries, where for i<j, A[i] > A[j]
- sorted array
```

Steps:

```
1. divide
2. sort and merge recursively, and count inversion pairs as merging
```

Run time: O(N Log N)
Implementation: A recursive method similar to MergeSort, but which I implemented via Python v.3.

```
In [39]: import random
         import time

         #generate a list of integer with random numbers
         def generate_list():
             size_ = int(input("insert size of the array: "))
             list_1 = [random.randrange(0, 101, 1) for _ in range(size_ + 1)]
             print("\n","Generated array is: ", "\n", list_1, "\n")
             return list_1
         #recursive divide
         def mergesort(list_):
             if len(list_) == 1:
                 return list_
             else:
                 m = len(list_) // 2
                 left_ = list_[:m]
                 right_= list_[m:]
                 return merge(mergesort(left_), mergesort(right_))
         #sort and count inversion as merging using an auxilary array
         def merge(left, right):
             sorted_ = []
```

```
            global count
            count = 0
            i = j = 0
            while i < len(left) and j < len(right):
                if left[i] < right[j]:
                    sorted_.append(left[i])
                    i += 1
                else:
                    sorted_.append(right[j])
                    count = count + (len(left) - i)
                    j += 1
            sorted_.extend(left[i:])
            sorted_.extend(right[j:])
            return sorted_

        if __name__ == '__main__':
            start = time.time()
            sorted_ = merge_sort(generate_list())
            end = time.time ()
            print ("Sorted array is: ", "\n",sorted_)
            print ("Number of inversions in the unsorted array are:", count)
            print ("time to compile= ", end- start)
```

```
insert size of the array: 200

 Generated array is:
 [69, 6, 48, 24, 99, 1, 5, 19, 80, 10, 11, 13, 4, 3, 10, 69, 96, 72, 63, 3, 36, 33, 63, 20, 90

Sorted array is:
 [0, 1, 1, 3, 3, 3, 3, 3, 4, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8, 8, 8, 9, 10, 10,
Number of inversions in the unsorted array are: 5009
time to compile=  1.036893367767334
```