Shashank K

SE21UCSE198

CSE 3

DAA Assignment -3

Goal : Comparing the number of times the values of max and min are over written

Without using divide and conquer method

Function :

Without divide and conquer

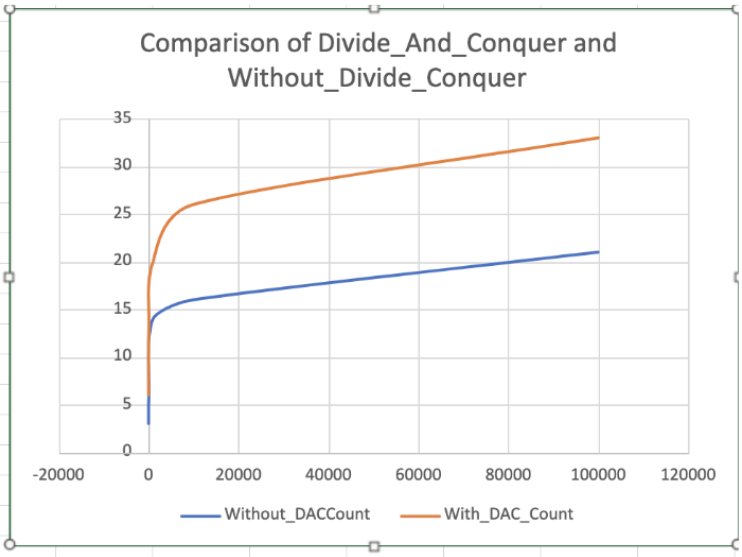
```
[ ]  time_without_divide = []
     updates_without = []
     def without(time_without_divide,updates_without,input_cases):
       for i in input_cases:
         start = time.time()
         min = i[0]
         max = i[0]
         update_min = 0
         update_max = 0
         for q in i:
           if min > q:
             min = q
             update_min+=1
           if q > max:
             max = q
             update_max+=1
           end = time.time()
         lis = [update_min,update_max]
         time_taken = end-start
         updates_without.append(lis)
         time_without_divide.append(time_taken)
       return time_without_divide,updates_without
```

With using divide and conquer method

```python
def findMinAndMax(nums, left, right, min=float("inf"), max=float("-inf"),min_count=0,max_count = 0):


    if left == right:

        if min > nums[right]:
            min = nums[right]
            min_count +=1

        if max < nums[left]:
            max = nums[left]
            max_count+=1

        return min, max,min_count,max_count


    if right - left == 1:

        if nums[left] < nums[right]:
            if min > nums[left]:
                min = nums[left]
                min_count+=1

            if max < nums[right]:
                max = nums[right]
                max_count +=1

        else:
            if min > nums[right]:
                min = nums[right]
                min_count+=1

            if max < nums[left]:
                max = nums[left]
                max_count+=1

        return min, max,min_count,max_count

    mid = (left + right) // 2

    min, max,min_count,max_count = findMinAndMax(nums, left, mid, min, max,min_count,max_count)

    min, max,min_count,max_count = findMinAndMax(nums, mid + 1, right, min_count, max_count)

    return min, max,min_count,max_count
```

Results:

input	Without_[	With_DAC_Count
10	3	6
100	6	13
1000	14	20
10000	16	26
100000	21	33



Conclusion

Seeing the number of updates, using without divide and conquer approach is much feasible and optimal than with using divide and conquer.

Considering space complexity, without_divide_and_conquer is much feasible than the with_divide_and_conquer, because with_divide_and_conquer using much more space than the without_divide_and_conquer.