Divide-and-Conquer

May 25, 2019

What is Divide-and-Conque?

- Divide: Splite the problem into several subproblem. (e.g. divide the entire array into half)
- Conquer: Solve the subproblem (recrusively) applying the same algorithm (e.g. compare each element for the subarray)
- Combine: Use subproblem results to derive a final result for the original problem (e.g. merge)

Questions:

Given a permutation $a_1, a_2, \cdots, a_n$ of the number $1, 2, \cdots, n$ find the number of pairs (a_i, a_k) such that $i < k$ and $a_i > a_k$
The method we are using is similar to the merge sort!!
Firstly, we split array into two almost equal pieces.
Then we recrusively solve each subproblem by keep split the subarray into two equally arrays until we hit the base case.
We need to change the merge part a little bit, the algorithm is shown below:

Algorithm 1 Merge and Count

```
1: i → 1; j → 1; r → 0; C → ∅
2: while i ≤ m and j ≤ n do
3:    if A[i] < B[j] then
4:       append A[i] to C
5:       i++
6:    else
7:       append B[j] to C
8:       j++
9:       r = r + m - i + 1
10:   end if
11: end while
12: return (C ,r)
```

Explanation for line #9:
Since we are merging two array together and want to find out missed match pair. Since we assume (by the recrusively call) two subarrays is sorted and we already know that there is r_l pairs in the left subarray and r_r pairs in the right part.
However, we need to consider those pairs that is across two arrays (i.e. the value of element in left subarray is greater than the value of element in the right subarray.)
Therefore, we do the merge count on those two subarrays.
If the element in $A[i]$(left array) is greater than $B[j]$ that means every element in left array after index i is greater than than $B[j]$ which is the number of mismatched pairs.
Since we have m elements in the left array and the current index is i, then we have $m-i+1$.
And we recrusively call this for each subproblem.