Advanced Programming Lab

HW 2 Sorting

1. What is time and space complexity of algorithm used in Part 1.

- Until all elements become zero, do sorting then update the array.
- For worst case: loop will run ∑ (n+k) times, where k = 1 to n-1.
  ∑ (n+k) = n+(n-1) + n+(n-2) + ……+ n+(1)
  So **time complexity is O(n^2).**
- For sorting, counting sort is used. Two temporary array of lenth N is used for that.
  So space complexity is O(n).

2. What is time and space complexity of algorithm used in Part 2

- For 3…n do : mergesort and update the array.
- For mergesort time complexity is O(nlogn).
  So **time complexity of part-2 is O(n^2logn).**
- For sorting, merge sort is used. Temporary array of lenth N is used for that.
  So Space complexity is O(n).

3. In Part-2, n = 1, 2, or 3 is a special case. Provide an example as to why this is special

and how did you handle this case in your program.

- For n=1, hi-5 is not possible. So h must be zero.
- For n=2, hi-5 must be same because it is mutual.
- For n=3, by using the concept of graph theory, number of hi-5s between three persons can be calculate. If these number is zero or positive then it is valid otherwise invalid.
- For example, (5,4,3) is sequence then
  (5+4-3)/2 = 3 is number of hi-5 between two people. Likewise we can calculate other two.