DC Assignment - 1

Rohan S
S20160010073 (UG - 3)

Instruction:

Python 3.x+ Version required.

1. Type the below command to run the first code, sasaki's (n-1) round algorithm.

   python3 sasaki.py

   The no of elements in the array can be altered in the main function by changing n, by default n = 10. The array is initialised in the worst case (inversely sorted manner i.e., descending order).

   Time complexity : O(n^2)
   The array takes (n-1) rounds to sort itself and there are no elements, so be amortised analysis it is O(n*(n-1)) = O(n^2)

   Space complexity : O(n)
   K be the space required for 1 node in the line network, then O(n * K). Here, all the space required for local computation is also included in the arbitrary constant K.
   This makes the space complexity O(n).

   Message Exchanges: (2*(n-1))*(n-1)
   In a purely distributed setup where each of this node communication overhaul will outshine the amortised O() analysis, therefore the reduction in one round saves 2(n-1) pairs of message exchanges which is very significant.
   For each round there will be 2*(n-1) message exchanges, so for n-1 rounds there will be (2*(n-1))*(n-1) message exchanges.

2. Type the below command to run the second code, alternate (n-1) round algorithm.

   python3 alternate.py

   Time complexity : O(n^2)

The array takes (n-1) rounds to sort itself and there are no elements, so be amortised analysis it is O(n*(n-1)) = O(n^2)

Space complexity : O(n)
K be the space required for 1 node in the line network, then O(n * K). Here, all the space required for local computation is also included in the arbitrary constant K.
This makes the space complexity O(n).

Message Exchanges: (2* ceil(n/3))*(n-1)
For each round there will be 2*ceil(n/3) message exchanges, so for n-1 rounds there will be (2* ceil(n/3))*(n-1) message exchanges.