Input Parameters

Use for all runs

- Default parameters are hardcoded in the Configuration class.

Parameters to vary

- Number of random neighbors (K): 3, 10, 20
- Number of iterations: 20, 200, 1000

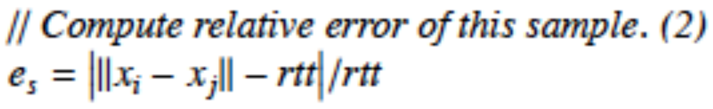
Results

For each simulation, please collect the following statistics on each node (functions to compute them are provided):

1. relative prediction error (between network distance and the rtt)
2. convergence of the algorithm solution by looking at the relative displacement of the nodes (i.e., how the nodes' positions vary in each iteration)

Hints

- For more details on Vivaldi, please refer to the following article http://pdos.csail.mit.edu/papers/vivaldi:sigcomm/paper.pdf
- Use the following definition of the relative prediction error (taken from the above paper)



```
// Compute relative error of this sample. (2)
$e_s = \left| \|x_i - x_j\| - rtt \right| / rtt$
```

What the report should contain

- State clearly any hypothesis you make
- For the metrics that are listed above under Results *you will have one value per node*. Please give for each metric
    - Mean, variance, maximum and minimum value (please use a table to present results)
    - Indicate the 50-th, 90-th and 99-th percentile of the relative predication error
    - A CDF (cumulative distribution function) or a CCDF of the values. The x-axis of each CDF is relative prediction error.
- Please discuss and comment the results you have obtained
- Be aware that the Internet delay space is not an Euclidian space, i.e. that the triangle inequality can be violated. Discuss how this fact impacts the Vivaldi algorithm.

- It is not necessary to print the Python program. However we will look at the online version you have turned in. So make sure that your code is
    - properly structured and contains **sufficient comments** so it is easily readable
    - executes properly