Consider that we have 10 base genes and 100 random population.

In genetic algorithm we have **cross over** and **mutation**.

Form those 100 population, 70% of them goes to cross over randomly, and the rest will go to mutation.

Cross Over: take two genes from the population, for example as follow:

AHTGSPFGJW

HTANDNWNTD

Then, we choose 30% of each gene randomly and do cross over:

AHTG – SPF – GJW --> AHTG – DNW – GJW --> AHTGDNWGJW

HTAN – DNW – NTD --> HTAN – SPF – NTD --> HTANSPFNTD

So, we will have 2 new genes out of that 2, and 4 genes in total.

So, from 70 of genes that will go to cross over, we will have 70 new genes and 140 genes in total.

Mutation: take one gene from the population, for example as follow:

AHTGSPFGJWHTANDNWNTD

Then, we select 50% of that randomly, and generate random letters and replace them.

Our example has length 20, so the 50% of that is 10:

AHT – GSPFGJWHTA – NDNWNTD --> AHT – PGQPGDBDMQ – NDNWNTD --> AHTPGQPGDBDMQNDNWNTD

Hence, 30 genes will go to mutation and we will have 30 new genes and those old 30. 60 in total.

After cross over and mutation of the 100 genes, we will have 200 genes. Then, we calculate the LCS of each gene with each of those 10 base genes. And calculate the root squared average for each gene by this formula:

```
def squared_average(A): # A is the array of LCS distances
    temp = np.sqrt(np.sum(np.power(A,2)))/A.shape[0]
    return temp
```

Then, we sort the 200 genes in descending order based on the root squared average and discard the 101 to 200, and keep just 1 to 100. Now, we have first generation. Then we can continue to next generations until we want.