# traveling salesperson

genetic algorithm

the code description

# calculate fitness function

i have an array of population that store the x,y of each city

and store them in array called city and store their there order in array called order

then i gave them fitness score and every population of i using function called calcdistance that takes two parameters (cities,population of i)

and fitness[i] =(1/(distance+1) and then make condition if(distence<recorddistence//recorddistence equal infinity)

then recorddistence=distance and bestever=population[i] and bestcurrent=population of i

# normalize fitness

after giving fitness for each population [i]

i will count the probability for each of them

and store them in fitness of i again

# pick one

roulette wheel part!!!

picking chromosome based on its probability

i made a variable and initialize it =1 the iam subtracting the probabilities of it and then shoes the last index before r=0 and return a population[that index]

# next generation

trying to make a new generation based on chromosome that has the bes probabilities

first i declared indexA and indexB and store the highest probability population[i] on them and then make a crossover that generates the offspring that i will mutate

and the result i will store it in new population array

and store the new population array in population array that iam trying to find the nest generation for till the loop ends

# crossover

applying the cross over on orderA,orderB without repetition

i have to store a random number from array of order in start variable

and choose another one from the same array but after the start and store it in end variable and add both of them in neworder variable.

in orderB i choose the numbers that doesnot belong to the neworder variable and push it to the neworder then i return this variable to next generation to mutate.
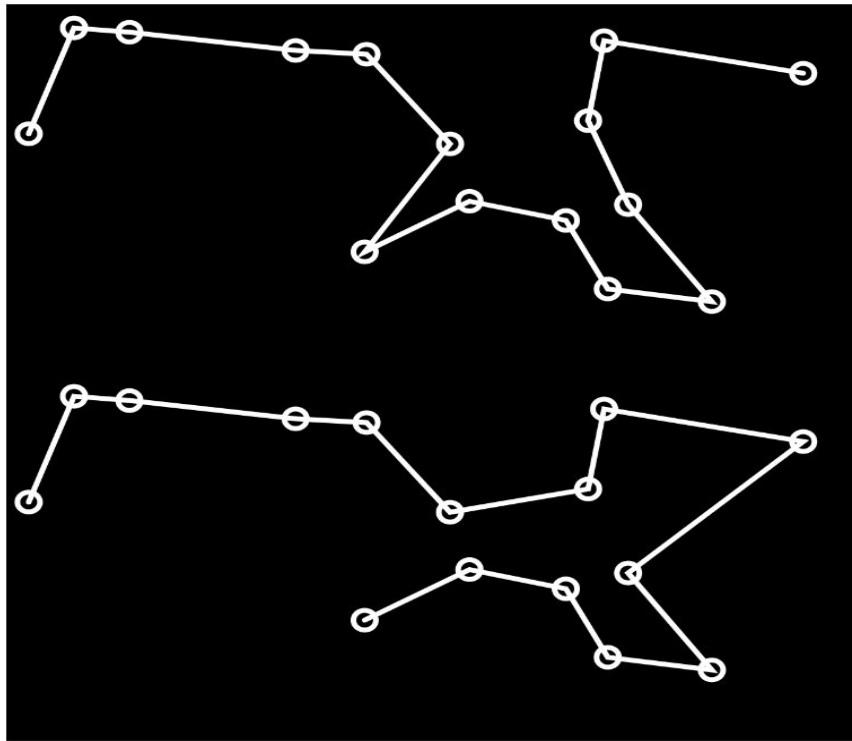
# mutation

in mutate function that takes one parameter //array of order

i choose two random numbers and swap them till the loop of numberofcities that describe the number of mutate repetition ends;

and then store order array in newpopulation ;

# code

https://editor.p5js.org/mahmoudrdwan/sketches/IXq-4LqBX