Operating Systems – 2
Programming Assignment –1 :Finding nearest point using multiple threads
Vanga Aravind Shounik – CS20TECH11055

Question:

Find the nearest point in a given set of points from a source in a two dimentional plane. In this program we are implementing a simpler version of the Travelling Salesperson Problem. Given a source point, (X,Y), and set of possible destinations, {(X1,Y1),(X2,Y2),(X3,Y3),...(Xn,Yn)}
Find the nearest point to (X,Y) among the given set using Euclidean distance formula.

Input:

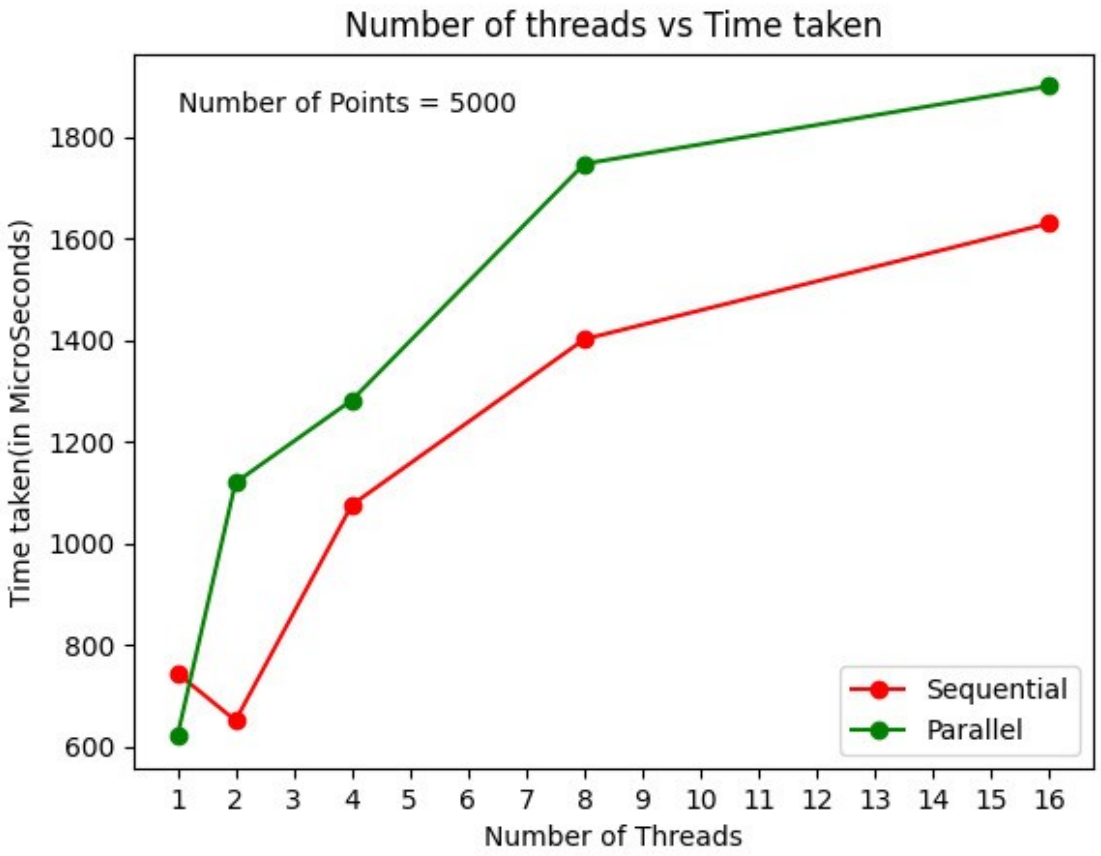
Here, the input should be in the form
number of threads
source point in the form (x,y)
number of points
all points in the form (x,y)

Implementation:

Here, the code takes the input from a text file named input.txt and then stores them in arrays of stores the points in structs which are named point and the number of threads and number of points are stored in integers. Then the structs thread attributes is created in which we store the attributes which should be sent to each thread. Then we divide the points to each thread and give the thread the array which contains the points and the head point and the number of points in the array. Then the thread is created and then the values are given in the form of Threadattr. Now, in the function, the thread attributes are saved in variables and then the point which is the nearest to the head point is returned by typecasting it into void pointer. Then the void pointer is recieved when the pthread is joined to the main thread and then the void pointer is typescasted back to struct point pointer. Now, the values returned are stored in another array named ThreadReturnPoints and now in a for loop, we find the point which is the nearest to the head point. Then we print the value int the terminal.

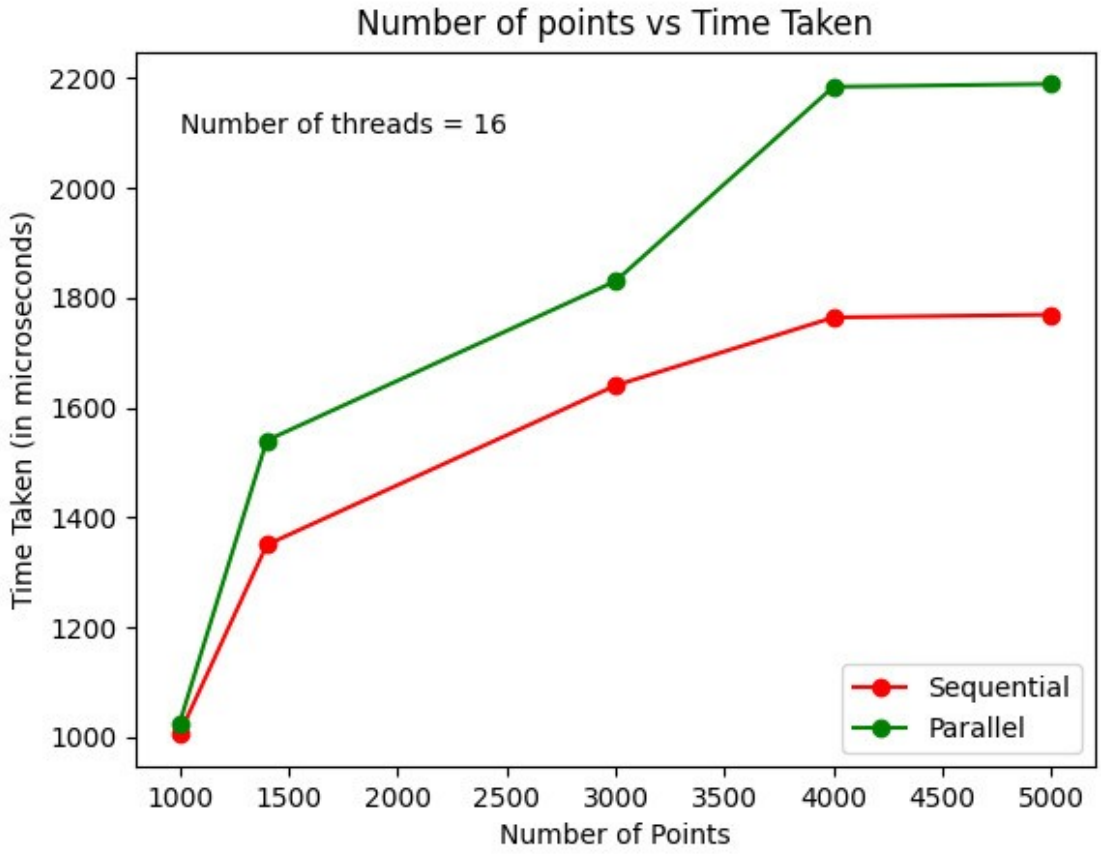
Analysis of the Results:

1. Varying number of threads: In this experiment, I varied the number of threads while keeping the total number of points fixed to 5000.



Here, we can see the difference between sequential and parallel execution of threads. Here, sequential is represented by the red line and parallel is represented by the green line.

2.Varying the input set size: In this experiment, I varied the number of points while keeping the total number of threads fixed to 16.



Here, we can see the difference between sequential and parallel execution of threads. Here, sequential is represented by the red line and parallel is represented by the green line.