

ASSIGNMENT 1

Topic: Parallelizing Traveling Salesman Problem with OpenMP

Student Name: Pooja Nadagouda

The main functions implemented as part of this program:

evaluate (), *crossover ()* and *mutate ()*

Also, there are couple of util functions implemented like calculating distance between coordinate points, computing index of the city and distance comparator. For sorting in built function is used.

Following functions are parallelized:

evaluate () and *crossover ()*

- The shortest distance obtained: **447.638**
- Time taken for computing shortest distance using 4 Thread: **11500695**

Performance calculation:

Elapsed time using 1 thread for Tsp (professor program) = 26372079

Elapsed time using 4 threads for Tsp (my program) = 11500695

Performance = $26650898 / 11500695 = 2.2$

Parallelization:

- **Evaluate function:**

In case of *evaluate()*, function is parallelized using OpenMP using the clause *#pragma omp parallel for*

Here there is no need of reduction clause as each thread calculates distance of each trip independently by summing up the distance between cities.

- **Crossover function:**

Similarly in case of *crossover()* function same parallelization clause is used.

The implementation strategy used here to reduce the computation time is as follows:

For computing ith child:

The first city in child[i] is the first city of parent[i]. For the second city in child[i], we compare second city of parent[i] and first city of parent[i+1], we choose the one that is not visited and is closer to the last appended city in child's trip. Then the pointer of parent[i] or parent[i+1] is incremented based on which is chosen.

- If both cities from parent[i] and parent[i+1] are not visited, we take the one with shortest distance and increment the pointer of parent from which city is chosen.
- If one of the cities from parent[i] and parent[i+1] is visited, then we take the city which is not visited and increment pointers of both parents since they are visited.

- If both cities from parent[i] and parent[i+1] are visited then we increment pointers of both parents and to next city in parent[i] and parent[i+1], then repeating the same strategy as above.

To keep track of visited cities an array of size 36 is created. The array has indexing for [A, B, C, D.....Y, Z, 0, 1,...8, 9] from 0-36 respectively. When a city is added to the child[i] we make the index of that city equal to 1.

To compute the child[i+1] an array of size 36 holding the complement is created. The array has indexing for [9, 8,...1, 0, Z, Y.....C, B, A] from 0-36 respectively. When a city is added to child[i], we fetch its complement from the above complement array and is appended to child[i+1]

- **Mutate function:**

Mutate function is not parallelized as it makes use of *rand()* function to generate random numbers. Parallelization of this increases the execution time.

Screenshots of TSP program executions:

Using 4 threads:

```
[[poojan26@cssmpi2h prog1]$ ./Tsp 4
```

```
# threads = 4
```

```
28836generation: 0
```

```
generation: 0 shortest distance = 1265.72
generation: 1 shortest distance = 1112.17
generation: 2 shortest distance = 1010.16
generation: 3 shortest distance = 886.145
generation: 4 shortest distance = 854.347
generation: 5 shortest distance = 754.303
generation: 6 shortest distance = 726.547
generation: 7 shortest distance = 650.497
generation: 8 shortest distance = 645.788
generation: 9 shortest distance = 623.044
generation: 10 shortest distance = 564.225
generation: 13 shortest distance = 544.999
generation: 14 shortest distance = 521.913
generation: 16 shortest distance = 503.681
generation: 17 shortest distance = 499.84
generation: 20
generation: 20 shortest distance = 491.064
generation: 22 shortest distance = 475.865
generation: 26 shortest distance = 465.408
generation: 28 shortest distance = 461.363
generation: 29 shortest distance = 456.864
generation: 36 shortest distance = 452.707
generation: 40
generation: 44 shortest distance = 449.552
generation: 51 shortest distance = 447.638
generation: 60
generation: 80
generation: 100
generation: 120
generation: 140
elapsed time = 11500695
```

```
itinerary = V1SPMBQAN26G4J37DX80TF95ZUH0EYRLCWKI
itinerary = W5IFDQ3ARPCS20BYEM08XTGKL7J4N96HUZV1
itinerary = V1JKXN4890YE0WSCPMBQ3DRA6U27LFIT5GHZ
itinerary = V1JA7KXN4890IYE0UWSCPMBQ3DRLF62GT5HZ
itinerary = UYVI06TXKJES5MPBQDR3L7GN9F4AHZ02WC18
itinerary = V6X98GFAL7R3DQBPM2UY1I04NKTS0JHZ5WCE
itinerary = V1YITN4KA06JUZE2WCS5MBDR3QP7F8XG9L0
itinerary = V1YI6TN4XK98GFAL7R3DQBPM2U0JZHEWSC05
itinerary = 1V6I084NXXFALJEUYZH0WSC25PBMQDR379TG
itinerary = V1I069FXNT48KGAL73RDPMMC20JUEHZYSBQ5
itinerary = V1I0684NTX9GKFLA7J0EUZYH2W5CMSBDR3QP
itinerary = V1YZHU0EJOI684NTXKFAL9G7R3DBQPSWC52M
itinerary = V1YZHUE0J6I048TNXK9FGA7L3RDQPM5WC52B
itinerary = V1YHZUJ60I84NTGXKF9AL7R3DBPM2E05CWSQ
itinerary = V1YZHUE0J60I84NTXFKG9AL7R3DBQPSWC52M
```

```
itinerary = V1YZHUE0J60I84NTGXKF9AL7R3DBPM25CWSQ
itinerary = V1YZHUE0JOI684NTXGK9FAL7R3DBQPM5WC2
itinerary = V1YZHUE20J60I84NTXKF9GAL7R3DQBPM5WC5
itinerary = V1YZHUE20J60I84NTXGK9FAL7R3DBQPM5WC5
itinerary = V1YZHUE20J60I84NTXGK9FAL7R3DQBPM5WC5
itinerary = V1YZHUE20J60I84NTXGK9FAL7R3DQBPM5WC5
```

```
itinerary = V1YZHUE20J60I84NTXGK9FAL7R3DBQPM5WC5
itinerary = V1YZHUE20J60I84NTXGK9FAL7R3DBQPM5WC5
```

Using 3 thread:

```
[poojan26@cssmpi2h prog1]$ ./Tsp 3
```

```
# threads = 3
28836generation: 0
generation: 0 shortest distance = 1265.72
generation: 1 shortest distance = 1112.17
generation: 2 shortest distance = 1010.16
generation: 3 shortest distance = 886.145
generation: 4 shortest distance = 854.347
generation: 5 shortest distance = 754.303
generation: 6 shortest distance = 726.547
generation: 7 shortest distance = 650.497
generation: 8 shortest distance = 645.788
generation: 9 shortest distance = 623.044
generation: 10 shortest distance = 564.225
generation: 13 shortest distance = 544.999
generation: 14 shortest distance = 521.913
generation: 16 shortest distance = 503.681
generation: 17 shortest distance = 499.84
generation: 20
generation: 20 shortest distance = 491.064
generation: 22 shortest distance = 475.865
generation: 26 shortest distance = 465.408
generation: 28 shortest distance = 461.363
generation: 29 shortest distance = 456.864
generation: 36 shortest distance = 452.707
generation: 40
generation: 44 shortest distance = 449.552
generation: 51 shortest distance = 447.638
generation: 60
generation: 80
generation: 100
generation: 120
generation: 140
elapsed time = 14539200
```

```
itinerary = V1SPMBQAN26G4J37DX80TF95ZUH0EYRLCWKI
itinerary = W5IFDQ3ARPCS20BYEM08XTGKL7J4N96HUZV1
itinerary = V1JKXN4890YE0WSCPMBQ3DRA6U27LFIT5GHZ
itinerary = V1JA7KXN4890IYE0UWSCPMBQ3DRLF62GT5HZ
itinerary = UYVIO6TXKJES5MPBQDR3L7GN9F4AHZ02WC18
itinerary = V6X98GFAL7R3DQBPM2UY1IO4NKTS0JHZ5WCE
itinerary = V1YITN4KA06JUZE2WCS5MBDR3QP7F8XG9L0
itinerary = V1YI6TN4XK98GFAL7R3DQBPM2U0JZHEWSC05
itinerary = 1V6IO84NXXKFALJEUYZH0WSC25PBMQDR379TG
itinerary = V1IO69FXNT48KGAL73RDPMMC20JUEHZYSBQ5
itinerary = V1IO684NTX9GKFLA7J0EUZYH2W5CMSBDR3QP
itinerary = V1YZHU0EJOI684NTXKFAL9G7R3DBQPMSWC52M
itinerary = V1YZHUE0J6IO48TNXK9FGA7L3RDQPMSWC52B
itinerary = V1YHZUJ6OI84NTGXXKF9AL7R3DBPM2E05CWSQ
itinerary = V1YZHUE0J6OI84NTXFKG9AL7R3DBQPMSWC52M
```

```
itinerary = V1YZHUE0J6OI84NTGXXKF9AL7R3DBPM25CWSQ
itinerary = V1YZHUE0J6OI84NTXGK9FAL7R3DBQPMSW5C2
itinerary = V1YZHUE20J6OI84NTXKF9GAL7R3DQBPMMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSW5C
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPMMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPMMSWC5
```

```
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSWC5
```

Using 2 threads:

```
[poojan26@cssmpi2h prog1]$ ./Tsp 2
```

```
# threads = 2
28836generation: 0
generation: 0 shortest distance = 1265.72
generation: 1 shortest distance = 1112.17
generation: 2 shortest distance = 1010.16
generation: 3 shortest distance = 886.145
generation: 4 shortest distance = 854.347
generation: 5 shortest distance = 754.303
generation: 6 shortest distance = 726.547
generation: 7 shortest distance = 650.497
generation: 8 shortest distance = 645.788
generation: 9 shortest distance = 623.044
generation: 10 shortest distance = 564.225
generation: 13 shortest distance = 544.999
generation: 14 shortest distance = 521.913
generation: 16 shortest distance = 503.681
generation: 17 shortest distance = 499.84
generation: 20
generation: 20 shortest distance = 491.064
generation: 22 shortest distance = 475.865
generation: 26 shortest distance = 465.408
generation: 28 shortest distance = 461.363
generation: 29 shortest distance = 456.864
generation: 36 shortest distance = 452.707
generation: 40
generation: 44 shortest distance = 449.552
generation: 51 shortest distance = 447.638
generation: 60
generation: 80
generation: 100
generation: 120
generation: 140
elapsed time = 14754190
```

```
itinerary = V1SPMBQAN26G4J37DX80TF95ZUH0EYRLCWKI
itinerary = W5IFDQ3ARPCS20BYEM08XTGKL7J4N96HUZV1
itinerary = V1JKXN4890YE0WSCPMBQ3DRA6U27LFIT5GHZ
itinerary = V1JA7KXN4890IYE0UWSCPMBQ3DRLF62GT5HZ
itinerary = UYVIO6TXKJES5MPBQDR3L7GN9F4AHZ02WC18
itinerary = V6X98GFAL7R3DQBPM2UY1IO4NKTS0JHZ5WCE
itinerary = V1YITN4KA06JUZE2WCS5MBDR3QP7F8XG9L0
itinerary = V1YI6TN4XK98GFAL7R3DQBPM2U0JZHEWSC05
itinerary = 1V6IO84NXXKFALJEUYZH0WSC25PBMQDR379TG
itinerary = V1IO69FXNT48KGAL73RDPMMC20JUEHZYSBQ5
itinerary = V1IO684NTX9GKFLA7J0EUZYH2W5CMSBDR3QP
itinerary = V1YZHU0EJOI684NTXKFAL9G7R3DBQPMSWC52M
itinerary = V1YZHUE0J6IO48TNXK9FGA7L3RDQPMSWC52B
itinerary = V1YHZUJ6OI84NTGXXKF9AL7R3DBPM2E05CWSQ
itinerary = V1YZHUE0J6OI84NTXFKG9AL7R3DBQPMSWC52M
```

```
itinerary = V1YZHUE0J6OI84NTGXXKF9AL7R3DBPM25CWSQ
itinerary = V1YZHUE0J6OI84NTXGK9FAL7R3DBQPMSW5C2
itinerary = V1YZHUE20J6OI84NTXKF9GAL7R3DQBPMMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSW5C
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPMMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPMMSWC5
```

```
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSWC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPMSWC5
```

Using 1 thread:

```
# threads = 1
28836generation: 0
generation: 0 shortest distance = 1265.72
generation: 1 shortest distance = 1112.17
generation: 2 shortest distance = 1010.16
generation: 3 shortest distance = 886.145
generation: 4 shortest distance = 854.347
generation: 5 shortest distance = 754.303
generation: 6 shortest distance = 726.547
generation: 7 shortest distance = 650.497
generation: 8 shortest distance = 645.788
generation: 9 shortest distance = 623.044
generation: 10 shortest distance = 564.225
generation: 13 shortest distance = 544.999
generation: 14 shortest distance = 521.913
generation: 16 shortest distance = 503.681
generation: 17 shortest distance = 499.84
generation: 20
generation: 20 shortest distance = 491.064
generation: 22 shortest distance = 475.865
generation: 26 shortest distance = 465.408
generation: 28 shortest distance = 461.363
generation: 29 shortest distance = 456.864
generation: 36 shortest distance = 452.707
generation: 40
generation: 44 shortest distance = 449.552
generation: 51 shortest distance = 447.638
generation: 60
generation: 80
generation: 100
generation: 120
generation: 140
elapsed time = 18209744
```

```
itinerary = V1SPMBQAN26G4J37DX80TF95ZUH0EYRLCWKI
itinerary = W5IFDQ3ARPCS20BYEM08XTGKL7J4N96HUZV1
itinerary = V1JKXN4890YE0WSCPMBQ3DRA6U27LFT5GHZ
itinerary = V1JA7KXN4890IYE0UWSCPMBQ3DRLF62GT5HZ
itinerary = UYVIO6TXKJES5MPBQDR3L7GN9F4AHZ02WC18
itinerary = V6X98GFAL7R3DQBPM2UY1I04NKTS0JHZ5WCE
itinerary = V1YITN4KA06JUZHE2WCS5MBDR3QP7F8XG9L0
itinerary = V1YI6TN4XK98GFAL7R3DQBPM2U0JZHEWSC05
itinerary = 1V6I084NXXKFALJEUYZH0WSC25PBMQDR379TG
itinerary = V1I069FXNT48KGAL73RDPMWC20JUEHZYSBQ5
itinerary = V1I0684NTX9GKFLA7J0EUZYH2W5CMSBDR3QP
itinerary = V1YZHU0EJOI684NTXKFAL9G7R3DBQPSWC52M
itinerary = V1YZHUE0J6I048TNXK9FGA7L3RDQPM5WC52B
itinerary = V1YHZUJ6OI84NTGXKF9AL7R3DBPM2E05CWSQ
itinerary = V1YZHUE0J6OI84NTXFKG9AL7R3DBQPSWC52M

itinerary = V1YZHUE0J6OI84NTGXKF9AL7R3DBPM25CWSQ
itinerary = V1YZHUE0JOI684NTXGK9FAL7R3DBQPM5WC2
itinerary = V1YZHUE20J6OI84NTXKF9GAL7R3DQBPM5WC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPM5WC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPM5WC5
itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DQBPM5WC5

itinerary = V1YZHUE20J6OI84NTXGK9FAL7R3DBQPM5WC5
itinerary = V1YZHUE20J6OI84TXGK9FAL7R3DBQPM5WC5
```

Using 1 thread – Professor's program

```
[[poojan26@cssmpi3h prog1]$ ./Tsp
usage: Tsp #threads
# threads = 1
generation: 0
generation: 0 shortest distance = 1265.72
generation: 1 shortest distance = 1083.52
generation: 2 shortest distance = 1009.03
generation: 3 shortest distance = 810.552
generation: 5 shortest distance = 806.414
generation: 6 shortest distance = 740.841
generation: 7 shortest distance = 680.579
generation: 8 shortest distance = 677.373
generation: 10 shortest distance = 660.018
generation: 11 shortest distance = 648.772
generation: 12 shortest distance = 538.668
generation: 15 shortest distance = 503.087
generation: 19 shortest distance = 491.081
generation: 20
generation: 21 shortest distance = 489
generation: 24 shortest distance = 471.777
generation: 25 shortest distance = 471.246
generation: 27 shortest distance = 470.528
generation: 36 shortest distance = 468.509
generation: 39 shortest distance = 466.045
generation: 40
generation: 50 shortest distance = 465.98
generation: 57 shortest distance = 462.728
generation: 60
generation: 61 shortest distance = 461.129
generation: 73 shortest distance = 460.2
generation: 76 shortest distance = 458.176
generation: 79 shortest distance = 457.506
generation: 80
generation: 83 shortest distance = 454.86
generation: 91 shortest distance = 452.975
generation: 100
generation: 113 shortest distance = 450.238
generation: 118 shortest distance = 449.658
generation: 120
generation: 140
elapsed time = 26372079
```

```
itinerary = V1SPMBQAN26G4J37DX80TF95ZUH0EYRLCWKI
itinerary = VG4XAK3R78TZMBW5H0EYU12DIN960JPCSQLF
itinerary = V120EYUJTZMPCSBW5HQLFG6OXAK3R7DIN489
itinerary = VYHE2SWMPQBC5D3RAN4I80J6XF9KL7GT1OUZ
itinerary = VIOFA7RDBQ3GT4KE2SWC5UZH0MP1Y689NXLJ
itinerary = V1YZH0DR3A7QBPM5WC58T4XKFNGL9E2JIOU6
itinerary = 1VIO84TF9NXGK6AL7R3DBMPS5C2U0EJQWYZH
itinerary = V1YZ0E2SWCHU3R7LF9KGNT48XJ6OI5MPBQDA
itinerary = V1ZYHUE2WCSMPBD3R7LF9KGNT48XJ6OI50AQ
itinerary = V1OI684NK90EUZH5CWSMPQBD37LAFGXJTY2
itinerary = V1YZH5CWS02EMPQBDR37LAKF984NTXGJU6OI
itinerary = V1YZHUE02WSC5MPQBDR37LA9FGXNT48J6OI
itinerary = V1YZH5CWSMPQBD37LAKFGXNT489J6OIEU02

itinerary = V1YZH5CWSMPBDR37LAFK94NTGX8IO6JUE02Q
itinerary = V1YZHUE025CWSMPQBDR37LAKF9GXNT486IOJ
itinerary = V1YZHUE02WSC5MPQBDR37LAFK9GXNT48IO6J
itinerary = V1YZHUE02WSC5MPQBDR37LA9FKGXNT48IO6J
itinerary = V1YZHUE02WSC5MPQBDR37LAFK9GXNT48IO6J
itinerary = V1YZHUE02WSC5MPQBDR37LAF9KGXNT48IO6J

itinerary = V1YZHUE025CWSMPQBDR37LAFK9GXNT480I6J
itinerary = V1YZHUE02WSC5MPQBDR37LAF9KGXNT48IO6J

itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J
itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J
itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J
itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J

itinerary = V1YZHUE025CWSMPQBDR37LAFK9GXNT48IO6J
itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J

itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT480I6J
itinerary = V1YZHUE025CWSMPQBDR37LAF9KGXNT48IO6J
```

Lab 1 Programs:

In case of pi_integral, for parallelization we use clause

#pragma omp parallel for private(x) reduction(+:sum)

Here private clause is used so that every thread has its own instance of x for calculations and reduction is used so that every thread updates the sum at then end of the loop using the updated value of sum.

In case of pi_monte, for parallelization we use clause

#pragma omp parallel firstprivate(x,y,radius,i) shared(count) reduction(+:pi)

Here firstprivate clause is used so that every thread begins with initialized value of x, y, i, radius for calculations. Shared clause is used so that there is only one instance of count (count of points in circle) and it is updated by one thread at a time. Reduction is used so that every thread uses the updated value of pi at the end of parallel program.

In case of pi_monte with parallelization, execution time increases as we use rand() function to generate random numbers. These random points are check if they fall within the circle or outside the circle.

Screenshots of pi_integral and pi_monte program executions:

Program - pi_integral_omp:

```
[poojan26@cssmpi8h lab1]$ ls
compile.sh pi_integral pi_integral.cpp pi_integral.o pi_integral_omp pi_monte pi_monte.cpp pi_monte.o pi_monte_omp Timer.cpp Timer.h Timer.o
[poojan26@cssmpi8h lab1]$ ./compile.sh
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 100000
Enter the number of threads: 4
elapsed time for pi = 772
# of trials = 100000, estimate of pi is 3.1415926535981269, Error is 0.0000000000083338
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 100000
Enter the number of threads: 3
elapsed time for pi = 872
# of trials = 100000, estimate of pi is 3.1415926535981384, Error is 0.0000000000083453
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 100000
Enter the number of threads: 2
elapsed time for pi = 1180
# of trials = 100000, estimate of pi is 3.1415926535981464, Error is 0.0000000000083533
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 100000
Enter the number of threads: 1
elapsed time for pi = 2108
# of trials = 100000, estimate of pi is 3.1415926535981615, Error is 0.0000000000083684
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 1000000
Enter the number of threads: 4
elapsed time for pi = 5599
# of trials = 1000000, estimate of pi is 3.1415926535898753, Error is 0.000000000000822
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 1000000
Enter the number of threads: 3
elapsed time for pi = 12784
# of trials = 1000000, estimate of pi is 3.1415926535899041, Error is 0.000000000001110
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 1000000
Enter the number of threads: 2
elapsed time for pi = 16608
# of trials = 1000000, estimate of pi is 3.1415926535898993, Error is 0.000000000001061
[poojan26@cssmpi8h lab1]$ ./pi_integral_omp
Enter the number of iterations used to estimate pi: 1000000
Enter the number of threads: 1
elapsed time for pi = 17239
# of trials = 1000000, estimate of pi is 3.1415926535897643, Error is 0.000000000000289
[poojan26@cssmpi8h lab1]$
```

Program – pi_monte_omp:

```
compile.sh pi_integral pi_integral.cpp pi_integral.o pi_integral_omp pi_monte pi_monte.cpp pi_monte.o pi_monte_omp Timer.cpp Timer.h Timer.o
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 10
Enter the number of threads: 1
elapsed time for pi = 21
# of trials = 10, estimate of pi is 2.7000000000000002, Error is 0.4415926535897929
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 10
Enter the number of threads: 2
elapsed time for pi = 190
# of trials = 10, estimate of pi is 6.2999999999999998, Error is 3.1584073464102067
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 10
Enter the number of threads: 3
elapsed time for pi = 187
# of trials = 10, estimate of pi is 8.1000000000000014, Error is 4.9584073464102083
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 10
Enter the number of threads: 4
elapsed time for pi = 243
# of trials = 10, estimate of pi is 12.300000000000007, Error is 9.1584073464102076
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 1000
Enter the number of threads: 1
elapsed time for pi = 153
# of trials = 1000, estimate of pi is 3.1440000000000001, Error is 0.0024073464102070
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 1000
Enter the number of threads: 2
elapsed time for pi = 6837
# of trials = 1000, estimate of pi is 7.4330000000000007, Error is 4.2914073464102076
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 1000
Enter the number of threads: 3
elapsed time for pi = 4408
# of trials = 1000, estimate of pi is 11.655999999999998, Error is 8.5144073464102057
[poojan26@cssmpi2h lab1]$ ./pi_monte_omp
Enter the number of iterations used to estimate pi: 1000
Enter the number of threads: 4
elapsed time for pi = 11435
# of trials = 1000, estimate of pi is 16.2980000000000018, Error is 13.1564073464102087
[poojan26@cssmpi2h lab1]$ █
```