Lab 10 report

6314061 – Quang Tuan Le

April 2023

Task: Random Graph, Serial Prim, Parallel Prim

- Here is the result between Serial and Parallel Prims (Same output)
- This is output of Serial Prim:

```
Edge   Weight
89 – 1 3
17 – 2 8
71 – 3 6
10 – 4 3
23 – 5 3
66 – 6 1
35 – 7 3
51 – 8 1
28 – 9 5
77 – 10       5
43 – 11       3
13 – 12       1
71 – 13       3
1 – 14 2
82 – 15       3
57 – 16       1
62 – 17       4
16 – 18       2
14 – 19       7
11 – 20       4
39 – 21       7
62 – 22       2
40 – 23       2
65 – 24       1
0 – 25 1
37 – 26       8
54 – 27       3
80 – 28       1
8 – 29 3
58 – 30       3
92 – 31       1
23 – 32       1
30 – 33       1
0 – 34 1
75 – 35       3
57 – 36       5
46 – 37       11
45 – 38       1
83 – 39       4
1 – 40 2
18 – 41       7
9 – 42 5
94 – 43       1
89 – 44       5
62 – 45       4
84 – 46       1
22 – 47       7
76 – 48       2
54 – 49       3
16 – 50       1
94 – 51       1
62 – 52       2
30 – 53       5
60 – 54       2
58 – 55       4
52 – 56       1
95 – 57       1
8 – 58 4
16 – 59       3
76 – 60       1
28 – 61       1
46 – 62       3
77 – 63       3
58 – 64       2
80 – 65       3
91 – 66       4
26 – 67       4
94 – 68       4
14 – 69       5
88 – 70       6
58 – 71       3
71 – 72       1
40 – 73       3
59 – 74       3
34 – 75       3
89 – 76       1
24 – 77       2
7 – 78 2
11 – 79       2
8 – 80 2
89 – 81       4
92 – 82       1
34 – 83       3
5 – 84 2
89 – 85       5
10 – 86       5
52 – 87       7
9 – 88 2
11 – 89       1
4 – 90 5
0 – 91 4
49 – 92       3
64 – 93       1
75 – 94       1
84 – 95       2
34 – 96       6
45 – 97       1
7 – 98 3
49 – 99       1
Serial   Time:   376
microseconds
```

- This is output of Parallel Prim:

```
Edge   Weight
89 – 1 3
17 – 2 8
71 – 3 6
10 – 4 3
23 – 5 3
66 – 6 1
35 – 7 3
51 – 8 1
28 – 9 5
77 – 10       5
43 – 11       3
13 – 12       1
71 – 13       3
1 – 14 2
82 – 15       3
57 – 16       1
62 – 17       4
16 – 18       2
14 – 19       7
11 – 20       4
39 – 21       7
62 – 22       2
40 – 23       2
65 – 24       1
0 – 25 1
37 – 26       8
54 – 27       3
80 – 28       1
8 – 29 3
58 – 30       3
92 – 31       1
23 – 32       1
30 – 33       1
0 – 34 1
75 – 35       3
57 – 36       5
46 – 37       11
45 – 38       1
83 – 39       4
1 – 40 2
18 – 41       7
9 – 42 5
94 – 43       1
89 – 44       5
62 – 45       4
84 – 46       1
22 – 47       7
76 – 48       2
54 – 49       3
16 – 50       1
94 – 51       1
62 – 52       2
30 – 53       5
60 – 54       2
58 – 55       4
52 – 56       1
95 – 57       1
8 – 58 4
16 – 59       3
76 – 60       1
28 – 61       1
46 – 62       3
77 – 63       3
58 – 64       2
80 – 65       3
91 – 66       4
26 – 67       4
94 – 68       4
14 – 69       5
88 – 70       6
58 – 71       3
71 – 72       1
40 – 73       3
59 – 74       3
34 – 75       3
89 – 76       1
24 – 77       2
7 – 78 2
11 – 79       2
8 – 80 2
89 – 81       4
92 – 82       1
34 – 83       3
5 – 84 2
89 – 85       5
10 – 86       5
52 – 87       7
9 – 88 2
11 – 89       1
4 – 90 5
0 – 91 4
49 – 92       3
64 – 93       1
75 – 94       1
84 – 95       2
34 – 96       6
45 – 97       1
7 – 98 3
49 – 99       1
Parallel Time:  253
microseconds
```

- I have tested on several times to measure the average time of each algorithm including Serial & Parallel:

Serial Prim	Parallel Prim
338	252
353	278
378	333
~ 356 (ms)	~ 288 (ms)

- The parallel prim works better than serial prim on a graph with 100 vertices.
- I use OpenMP to parallel at 2 tasks: First task is parallel to initialize all keys as infinite, and the second one is to pick minimum key vertex from the set of vertices not yet included in MST, and update all adjacent vertices of that vertex.