Short Project 9 Report

CS 6301.011

Savan Amitbhai Visalpara (sxv180069)
Srikumar Ramaswamy (sxr170016)

This report provides details on run time of our implementation of Boruvka's and Prim's algorithm for minimum spanning tree, and also compares their performance.

We ran our program on test cases provided for SP-9 as well as on random graphs as suggested.

Below table shows performance on test cases provided for SP-9.

No of V, E	**Boruvka's**	**Prim's**
50, 140 (mst-50-140-84590.txt)	Boruvka 84950 Time: 12 msec. Memory: 3 MB / 256 MB.	Prim IndexedHeap 84950 Time: 12 msec. Memory: 3 MB / 256 MB.
200, 580 (mst-200-580-153534.txt)	Boruvka 153534 Time: 19 msec. Memory: 4 MB / 256 MB.	Prim IndexedHeap 153534 Time: 13 msec. Memory: 4 MB / 256 MB.
10k, 30k (mst-10k-30k-1085305.txt)	Boruvka 1085305 Time: 127 msec. Memory: 57 MB / 256 MB.	Prim IndexedHeap 1085305 Time: 41 msec. Memory: 51 MB / 256 MB.

As we can see, for smaller cases both algorithm's running time is very similar. However, as the number grows, Prim's is clearly more efficient than the Boruvka's.

Below table compares the performance on randomly generated graph as suggested in the assignment.

No of V, E	Boruvka's	Prim's
10k, 1M	Boruvka 203410 Time: 462 msec. Memory: 350 MB / 1024 MB.	Prim IndexedHeap 203410 Time: 287 msec. Memory: 627 MB / 1024 MB.
100k, 30M	Boruvka 2147868221 Time: 7386 msec. Memory: 2800 MB / 4066 MB.	Prim IndexedHeap 2147868221 Time: 4578 msec. Memory: 2902 MB / 4066 MB.
100k, 300M	Boruvka 2147588463 Time: 81246 msec. Memory: 25545 MB / 256 26327.	Prim IndexedHeap 2147588463 Time: 45607 msec. Memory: 25545 MB / 26327 MB.
1M, 300M	Boruvka 2151331035 Time: 12850 msec. Memory: 25949 MB / 26327 MB.	Prim IndexedHeap 2151331035 Time: 46280 msec. Memory: 25949 MB / 26327 MB.

As we can see, Prim's algorithm is way more efficient than the Boruvka's. Note that we couldn't run 300M test cases for Boruvka's both on local machine and csgrad machine as it was throwing out of memory error. So, we extrapolated those values from known value.