Mandatory Project – 3:

Net Id: kxk152430

Shortest Algorithms, MST, Heap Sort and TSP using binary heaps, indexed heaps and priority queue

Prim's algorithms comparison of Edges and Vertices version

Instructions :
compile : javac -Xlint Driver.java
run : java MST (g1.txt/g2.txt/g3.txt/g4-big.txt)

g1.txt:

PrimMST-Edge weight :: 84950
Time: 2 msec.
Memory: 1 MB / 119 MB.

g2.txt:

PrimMST-Edge weight :: 110419
Time: 4 msec.
Memory: 2 MB / 119 MB.

g3.txt:

PrimMST-Edge weight :: 153534
Time: 4 msec.
Memory: 3 MB / 119 MB.

g4-big.txt:

PrimMST-Edge weight :: 10000
Time: 7288 msec.
Memory: 623 MB / 900 MB.

Time and memory analysis of Prim's MST with Edge and Vertices:

	Prim's MST With Edges		Prim's MST With Vertices	
	Time(msec)	Memory	Time(msec)	Memory
g1.txt	2	1 MB / 119 MB	5	1 MB /119MB
g2.txt	4	2 MB / 119 MB	7	2 MB /119MB
g3.txt	4	3 MB / 119 MB	8	3 MB / 119 MB
g4-big.txt	7288	623 MB / 900 MB	1688	365 MB / 921 MB

The Edge flavor of Prim's algorithm is better when the inputs are lesser than Million inputs like 100s, 1000s, additional reason for the performance is it is build with JAVA's priority queue.
When the input size is increased to Millions, then the Prim's algorithm with vertices implemented using Indexed Heaps performs better.

Traveling Salesman Problem

TSP algorithm gives correct result when the given input graph is a complete graph (a graph with edges between all vertices).
In order to run the TSP use the below instruction,

Instructions :

```
compile : javac -Xlint Driver.java
run : java Driver (g1.txt/g2.txt/g3.txt/g4-big.txt) 2
```

Outputs:

g1.txt :

```
TSP path :: 1 38 26 36 29 25 17 48 47 28 7 44 3 15 18 21 49 5 8 46 43 12 11 22 32 9 40 41 50 35 24 14
27 31 10 20 19 13 39 30 2 45 6 16 37 33 42 23 4 34
Time: 5 msec.
Memory: 1 MB / 119 MB.
```

g2.txt :

```
TSP path :: 1 30 98 64 79 47 37 48 95 41 78 36 63 46 8 13 6 32 15 59 44 72 65 25 81 84 22 70 23 33
82 11 80 31 100 38 86 42 75 51 92 60 85 76 24 52 61 55 27 88 2 43 74 89 69 18 54 29 4 49 5 21 40 67
71 66 28 90 19 97 58 73 94 12 17 10 53 91 83 50 7 68 87 93 39 20 9 14 45 35 34 3 26 77 96 99 16 62
57 56
Time: 7 msec.
Memory: 2 MB / 119 MB.
```

g3.txt :

```
TSP path :: 1 115 171 105 104 6 71 16 185 154 133 49 8 33 10 20 136 108 132 86 55 163 66 150 90
174 70 37 124 21 181 63 146 196 74 50 167 18 2 35 40 130 121 29 192 200 41 75 137 107 156 19 54
76 148 93 140 187 78 165 99 149 84 117 153 101 43 96 125 38 118 91 24 3 129 22 179 103 183 57
197 34 166 199 191 4 159 188 161 95 25 36 180 109 39 134 61 11 116 80 45 12 82 52 164 139 126 87
85 13 42 173 5 15 77 89 88 170 176 31 158 142 198 60 114 59 175 9 72 64 17 68 177 28 94 143 102
113 127 195 48 162 7 190 112 169 27 53 14 193 51 69 32 79 92 100 81 184 30 98 56 119 58 47 67 172
44 135 62 186 110 151 83 128 182 157 122 155 141 168 152 144 194 73 46 131 65 189 138 97 123
111 160 147 23 178 120 26 106 145
Time: 17 msec.
```

```
Memory: 3 MB / 119 MB.

g4-big.txt :

TSP path :: 1 2 3 813 26 51 81 209 227 266 315 415 432 444 472 598 601 610 657 716 9089 8082
6385 6560 7490 6184 3088 5648 5553 2803 9903 3637 1129 4497 2756 9655 5703 7444 4352 6742
3950    …
1750 1503 2344 4638 9904 3598 1219 7078 1165 8032 3212 7735 1289 8012 1437
Time: 912 msec.
Memory: 348 MB / 917 MB.
```