Experimental results

runMSClient:

```
[java] apr 09, 2015 4:47:38 PM clients.Client runTask
[java] INFO: Task tasks.TaskMandelbrotSet@25f38edcTask time: 333 ms.
[java] apr 09, 2015 4:47:38 PM clients.Client end
[java] INFO: Client time: 498 ms.
```

runTSPClient:

```
[java] apr 09, 2015 4:45:22 PM clients.Client runTask
[java] INFO: Task tasks.TaskEuclideanTsp@1a86f2f1Task time: 22 ms.
[java] apr 09, 2015 4:45:22 PM clients.ClientEuclideanTsp getLabel
[java] INFO: Tour: 0 7 9 2 6 5 4 3 8 1
[java] apr 09, 2015 4:45:22 PM clients.Client end
[java] INFO: Client time: 227 ms.
```

Server	**Client**	MandelbrotSet runtime client [ms]	TSP runtime client [ms]
localhost	*localhost*	510	239
hobbes.cs.ucsb.edu	*own computer*	498	227 / 15429
Average		504	233

We implemented the TSP first with a bad, greedy algorithm (nearest neighbor) to just make things work for that particular problem. When experimenting with the permutation algorithm, we experienced a much longer runtime, but the results might be more correct.