Table 1-1

	Points within a Circle Runtime (ms)		
n*	Graham Scan	Jarvis March	Quickhull
10	0.037	0.22	0.062
1000	1.892	1.488	0.292
10000	5.042	5.426	0.652
100000	48.223	89.571	5.873
1000000	456.719	1129.02	59.062

Table 1-2

	Points On a Circle Runtime (ms)		
n*	Graham Scan	Jarvis March	Quickhull
10	0.036	0.029	0.067
1000	1.583	5.193	0.294
10000	3.034	31.865	0.669
100000	32.74	296.737	5.8
1000000	383.964	3110.07	59.301

Table 1-3

	Points within a Rectangle Runtime (ms)		
n*	Graham Scan	Jarvis March	Quickhull
10	0.037	0.024	0.065
1000	1.863	1.053	0.297
10000	5.112	3.731	0.604
100000	51.45	37.514	5.836
1000000	469.706	378.508	59.041

Table 1-4

	Points within a Triangle Runtime (ms)		
n*	Graham Scan	Jarvis March	Quickhull
10	0.037	0.023	0.069
1000	1.81	0.918	0.293
10000	4.691	1.476	0.634
100000	45.758	11.022	5.816
1000000	448.28	83.268	58.866

Runtime Complexity

	Running Time Complexity		
	Graham Scan	Jarvis March	Quickhull
Best Case	O(nlogn)	O(nlogn)	O(nlogn)
Average Case	O(nlogn)	O(nlogn)	O(nlogn)
Worst Case	O(nlogn)	O(n^2)	O(n^2)

For the most part, the empirical analysis corresponded to the theoretical analysis. For the case of points being within a circle and on a circle, it makes sense that Jarvis March was slow because it would have to check a lot of points before it found the point with the smallest angle. Jarvis march is fast on a rectangle and triangle because the points are closer together and it is easier to determine the smallest angle. Graham scan performed best on a circle and a little slower on everything else. This makes sense because the runtime is constant between best case, average case, and worst case. Quickhull had similar performance on each case.