Willia Hardy
Algorithms Project 2

(a) Complete the following 4 tables:

Table 1-1 Running time in millisecond for case 1 (points are within a circle)

n*	Running time		
	Graham Scan	Jarvis March	Quickhull
10	128	296	148
1000	1972	1366	5209
10,000	3788	3575	19342
100,000	36979	63952	282456
1,000,000	329254	904564	4.11618*E06

Table 1-2 Running time in millisecond for case 2 (points are on a circle)

n*	Running time		
	Graham Scan	Jarvis March	Quickhull
10	256	319	96
1000	1777	5073	11255
10,000	2606	22561	78882
100,000	24002	213659	699093
1,000,000	263075	2.03196*E06	6.92198*E06

Table 1-3 Running time in millisecond for case 3 (points are within a rectangle)

n*	Running time		
	Graham Scan	Jarvis March	Quickhull
10	312	305	141
1000	1956	763	3840
10,000	4910	3500	10448
100,000	40053	28085	48499
1,000,000	339838	249889	329651

Table 1-4 Running time in millisecond for case 4 (points are within a triangle)

n*	Running time		
	Graham Scan	Jarvis March	Quickhull
10	229	270	166
1000	1331	582	1542
10,000	7056	5393	8524
100,000	36849	11700	42567
1,000,000	321783	107412	130103

(b) What's the asymptotic time complexity of the three algorithms? Complete the following table:

	Running time complexity		
	Graham Scan	Jarvis March	Quickhull
Best case	n	nh	$n \log n$
Average case	$n \log n$	nh	$n \log n$
Worst case	$n \log n$	n^2	n^2

(c) Does your empirical analysis match with your theoretical analysis? Justify your answer.

Graham Scan had the best overall running time for all the datasets which correlates with its known average running time complexity of *nlogn*. However, when it came to within a rectangle and a triangle, the Jarvis March has the best running time. This makes sense because points on a rectangle and a triangle are already on a line and the run time is more linear than logarithmic. Quick Hull has the slowest running time in each case. I believe that this is due to the time it take to divide and conquer the convex instead of simply finding points like Graham Scan and Jarvis March.