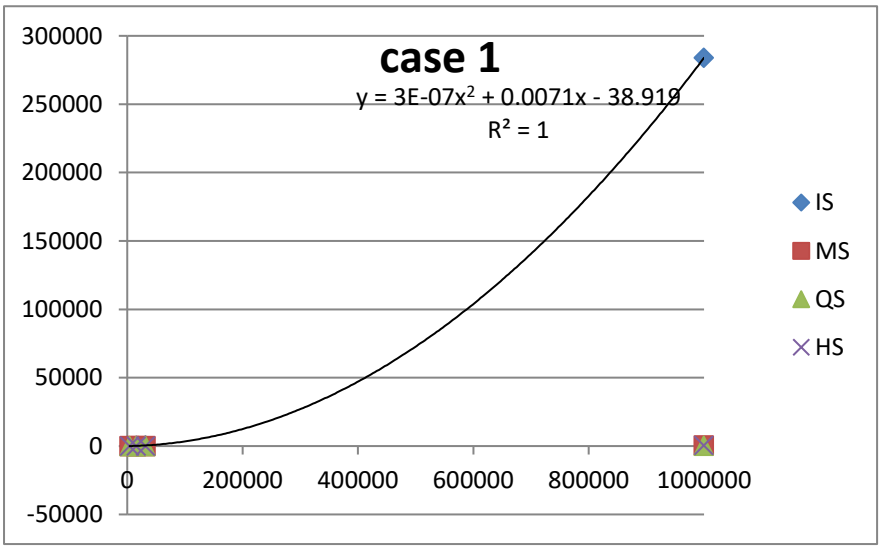
楊學翰 B08901054 演算法 PA1

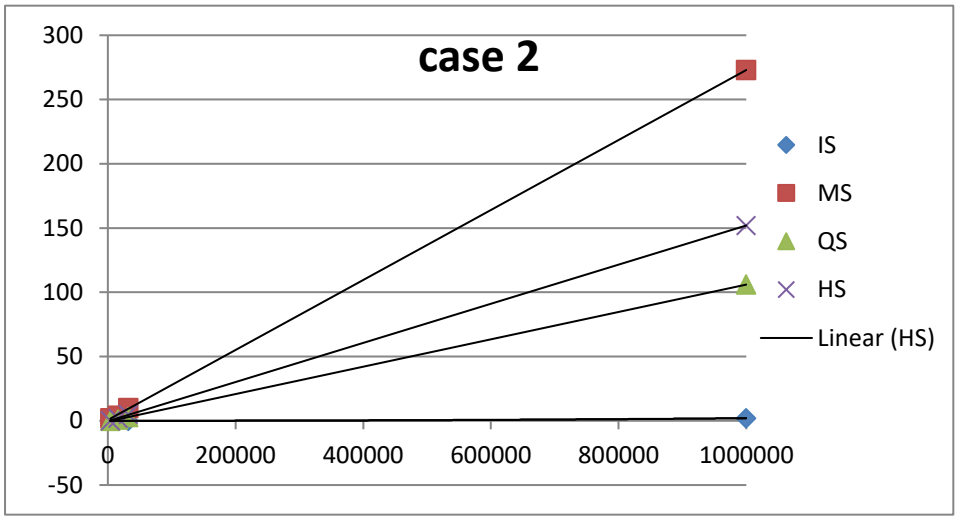
Input size	IS		MS		QS		HS	
	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)
4000.case2	0	12500	2	12500	0	12500	0	12500
4000.case3	15.998	12500	1	12500	0	12500	1	12500
4000.case1	7.999	12500	1	12500	1	12500	1	12500
16000.case2	0	12648	4	12648	1	12648	2	12648
16000.case3	150.977	12648	4.999	12648	1	12648	2	12648
16000.case1	119.982	12648	5.999	12648	2	12648	2	12648
32000.case2	0	12648	9.998	12648	2.999	12648	4.999	12648
32000.case3	774.883	12648	8.999	12648	3	12648	4.999	12648
32000.case1	481.927	12648	12.988	12648	4.999	12648	5.999	12648
1000000.case2	2	18668	272.959	20524	105.984	18668	151.977	18668
1000000.case3	568156	18668	304.953	20524	109.983	18668	178.973	18668
1000000.case1	283963	18668	445.932	20524	206.968	18668	291.956	18668

Case 1: Average case



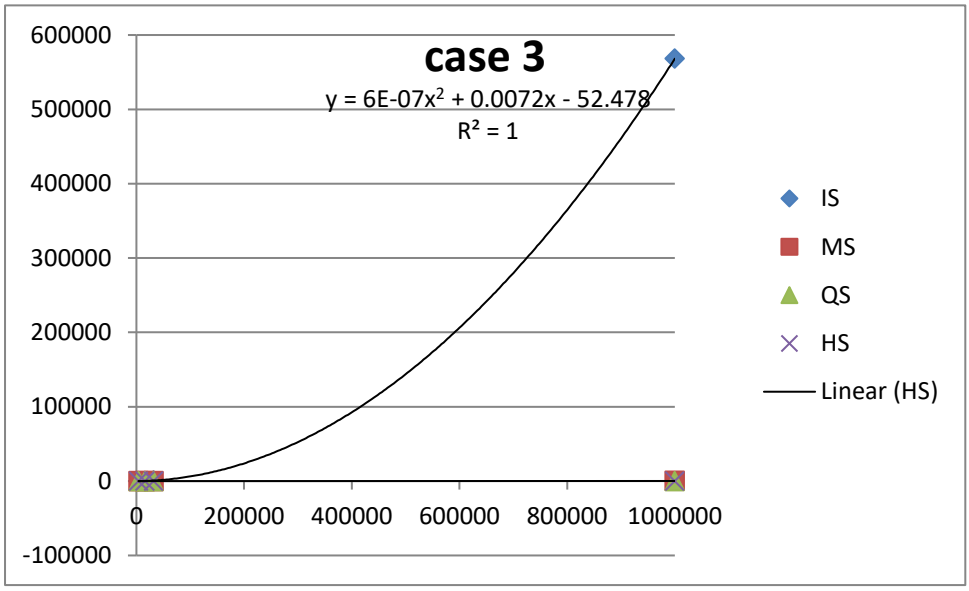
Insertion sort is $\theta(n^2)$, heap sort & merge sort are $\theta(n\log n)$, and quick sort is expected to be $\theta(n\log n)$

Case 2: Best case



Insertion sort is Ω(n), heap sort & merge sort are θ(nlogn), and quick sort is expected to be θ(nlogn)

Case 3: Worst case



Insertion sort is $\theta(n^2)$, heap sort & merge sort are θ(nlogn), and quick sort is $O(n^2)$, while randomized quick sort is expected to be θ(nlogn). I used randomized quick sort in this PA.