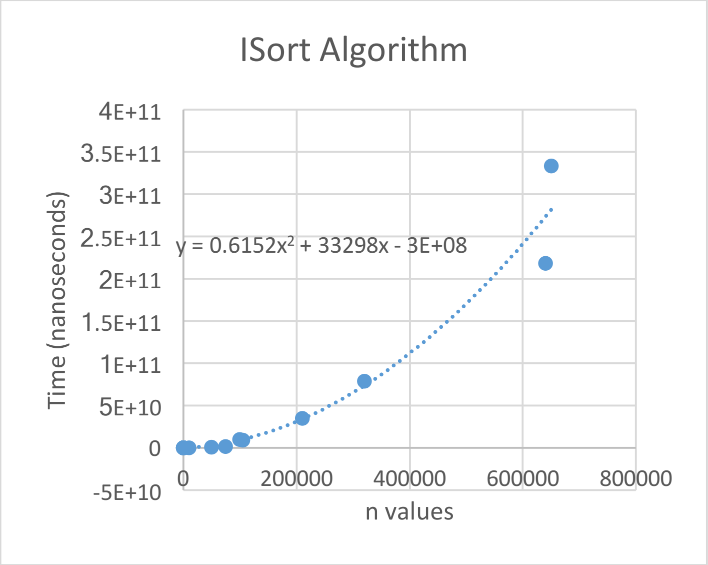
Questions 3 and 4

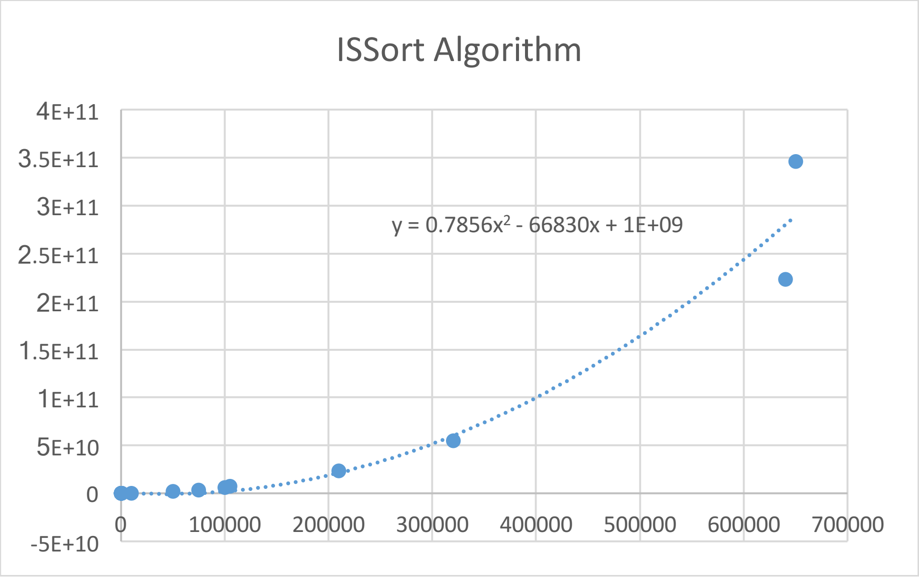
The ISort program has a Θ of n^2. The data from the table graph below, gives an equation of $y = 0.6152x^2 + 3329x - 3^{-8}$. The asymptotic analysis would suggest looking at the term in the x^2 and this suggests that the average run of $0.6152n^2$. The operation cost would then be the $3329n - 3^{-8}$.

ISort			
#	n	Times run (ns)	Average run (ns)
1	5	144284	28856.8
2	10	255361	25536.1
3	20	465545	23277.3
4	25	1058274	42331.0
5	50	3582155	71643.1
6	75	2390168	31868.9
7	150	5385962	35906.4
8	300	3038784	10129.3
9	400	5240031	13100.1
10	500	23189046	46378.1
11	1000	15171675	15171.7
12	10000	125066890	12506.7
13	50000	624598310	12492.0
14	75000	1401970517	18692.9
15	100000	9766784240	97667.8
16	105000	9141822991	87065.0
17	210000	34380956067	163718.8
18	320000	78607523770	245648.5
19	640000	2.17754E+11	340241.3
20	650000	3.33474E+11	513037.6



The table and graph below are for the ISSort algorithm.

ISSort			
#	n	Times run (ns)	Average run (ns)
1	5	3458185	691637.0
2	10	1056124	105612.4
3	20	1211014	60550.7
4	25	1672874	66915.0
5	50	1942275	38845.5
6	75	2615755	34876.7
7	150	4505205	30034.7
8	300	5193932	17313.1
9	400	9552553	23881.4
10	500	10491188	20982.4
11	1000	26452572	26452.6
12	10000	127846537	12784.7
13	50000	2000217307	40004.3
14	75000	3738761269	49850.2
15	100000	6235776669	62357.8
16	105000	7501073280	71438.8
17	210000	23874232394	113686.8
18	320000	54682324918	170882.3
19	640000	2.22811E+11	348141.9
20	650000	3.45615E+11	531715.3



From these, it can be seen that the average run is $0.7856n^2$ and the operation cost is $66830n - 1^9$.

Question 5

The structured Insertion Sort made things worse as it increased the run-time and the number of comparisons. Despite putting the descending runs in ascending order, the insertion sort still proceeded to check each number one by one in the unsorted part of the array, which renders the structured part redundant.