

BubbleSort

Command Line Argument	Time
10 ba bd br	ba 10 = 6.858399137854576e-05
	bd 10 = 6.806262535974383e-05
	br 10 = 0.0005890419706702232
50 ba bd br	ba =50 0.008313312486279756
	bd= 50 0.0010048539843410254
	br =50 0.006878500513266772
100 ba bd bn	ba 100 =0.00028383301105350256
	bd 100 = 0.006390770489815623
	bn 100 = 0.001791957940440625
500 ba bd br	ba 500 = 0.0032970209722407162
	bd 500 = 0.07595510449027643
	br 500 = 0.07886410440551117

InsertionSort

Command Line Argument	Time
10 ia id ir	ia 10 = 7.633341010659933e-05
	id 10 = 5.5896525736898184e-05
	ir 10 = 0.0005579375429078937
50 ia id ir	ia 50 = 0.00013097998453304172
	id 50 = 0.0005941879935562611
	ir 50 = 0.005907853483222425
100 ia id in	ia 100 = 0.0002062080311588943
	id 100 = 0.00414477102458477
	in 100 = 0.0010170619934797287
500 ba bd br	ia 500 = 0.0009124369826167822
	id 500 = 0.046107999922242016
	ir 500 = 0.05080268805613741

SelectionSort

Command Line Argument	Time
10 sa sd sr	sa 10 = 8.022901602089405e-05
	sd 10 = 4.8812536988407373e-05
	sr 10 = 0.001628395461011678
50 sa sd sr	sa 50 = 0.00034768745535984635
	sd 50 = 0.00033218745375052094
	sr 50 = 0.003020333591848612
100 sa sd sn	sa 100 = 0.000939957972150296
	sd 100 = 0.0024794795317575336
	sn 100 = 0.0019804369658231735
500 sa sd sr	sa 500 = 0.05042362544918433
	sd 500 = 0.023551916994620115
	sr 500 = 0.01977133355103433

When comparing the sorting algorithms, it is evident that when sorting large numbers selection sort is the fastest algorithm and then its insertion sort and then bubble sort. This is apparent through the 500 example shown within the table above. We can see from above that bubble sort goes from $N(O)$ to $N(O)^2$ as with more numbers used the time taken is increased. However the other algorithms stay consistent with $N(O)^2$ with selection sort being the best.