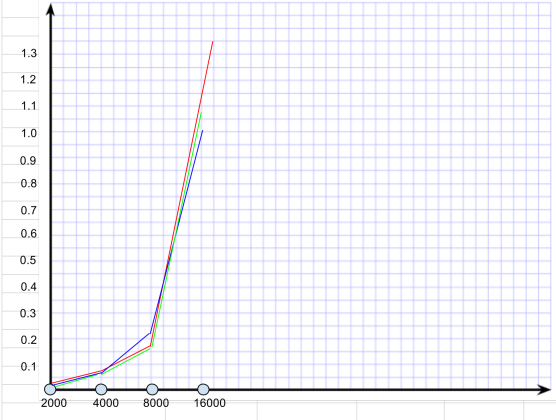
Question 2

TRY 1	2000	4000	8000	16000
movie_title_score.random	0.03	0.06	0.254	1.28
movie_title_score.reversed	0.021	0.055	0.222	0.984
movie_title_score.sorted	0.025	0.039	0.141	1.122

TRY 2	2000	4000	8000	16000
movie_title_score.random	0.028	0.062	0.246	1.304
movie_title_score.reversed	0.027	0.063	0.221	1.012
movie_title_score.sorted	0.027	0.039	0.135	1.098

TRY 3	2000	4000	8000	16000
movie_title_score.random	0.03	0.059	0.251	1.493
movie_title_score.reversed	0.03	0.055	0.22	1.022
movie_title_score.sorted	0.026	0.065	0.214	1.013

AVERAGE	2000	4000	8000	16000
random	0.02933333333	0.06033333333	0.18775	1.359
reversed	0.026	0.05766666667	0.221	1.006
sorted	0.026	0.04766666667	0.1633333333	1.077666667



Question 3

In accordance with the doubling hypothesis, when the input doubles, the runtime doubles. It's a linear relationship.

Question 4

Assuming that calling the "less" function doesn't add to the runtime, according to the cost model, since there's two array access in the sort method, the rate of growth would be ~ N ^ 2.

Question 5

By looking at the results of the graph paired with the code and the algorithm, since the exch function is called outside the second for loop, and since the less function is in the if statement, it does not matter the order of the input, the runtime will be similar (the same with a negligable difference).