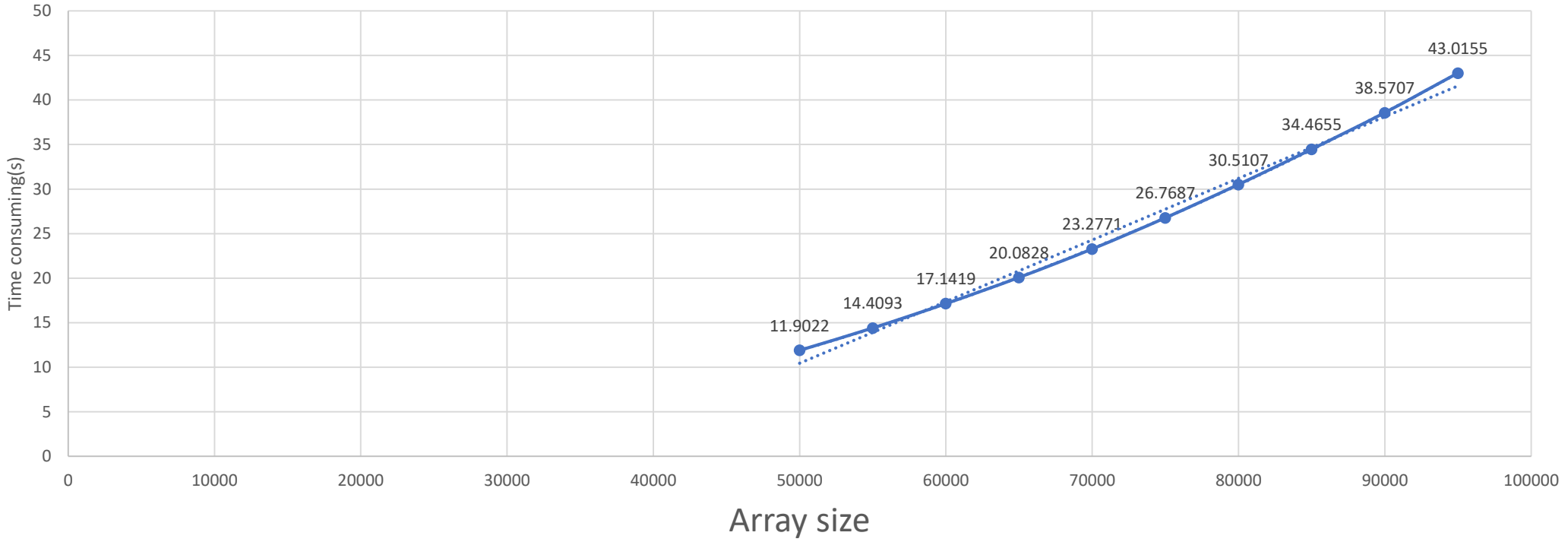
Lab07: Sort Analysis

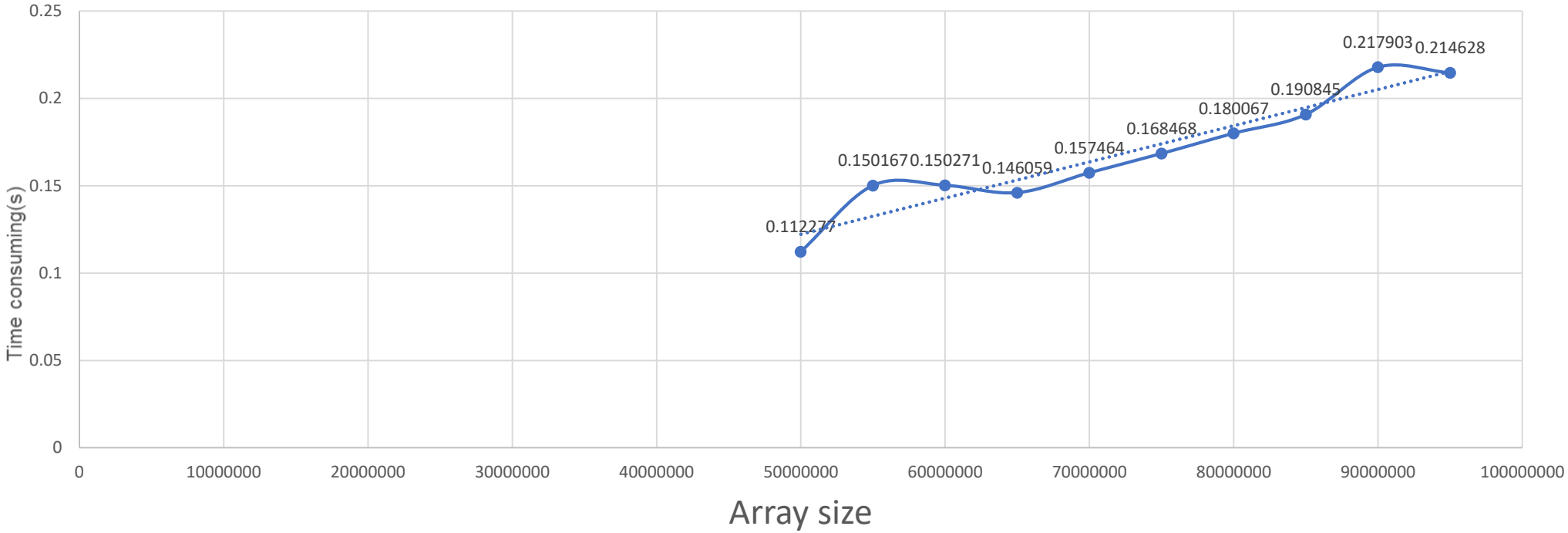
Guanyu Li

Bubble Sort: Random



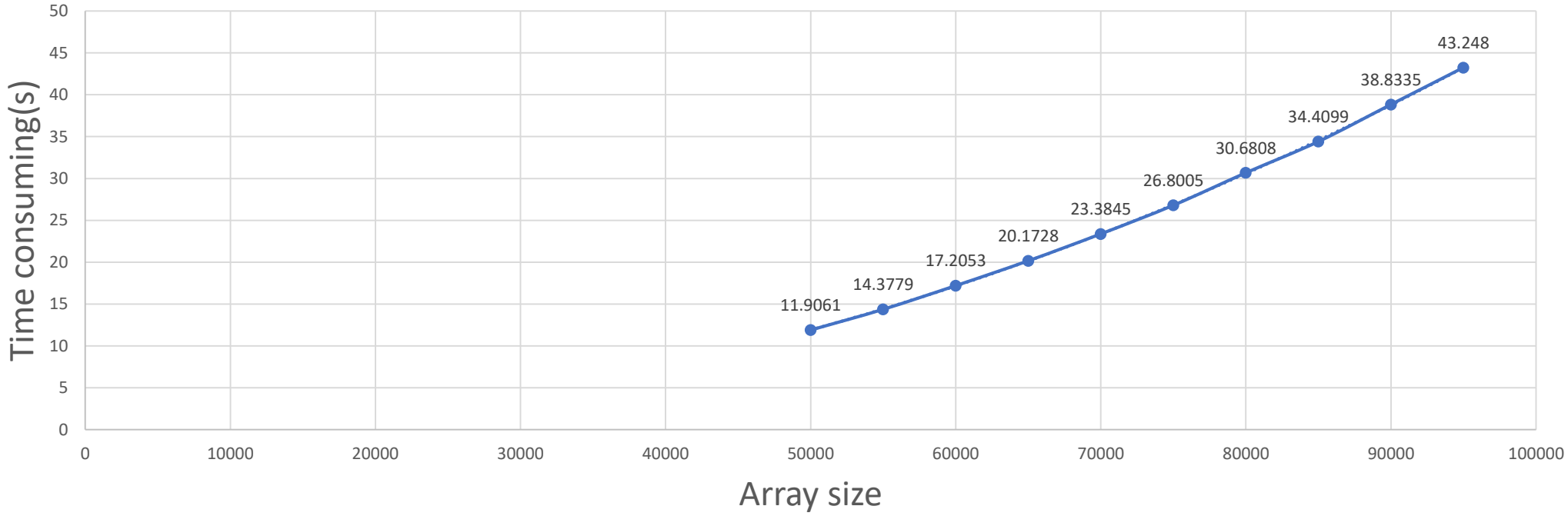
- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=1 in polynominal function.
- I think it will take 499100(s), since the trending line is close to a function:$y = 5E\text{-}09x^2 - 9E\text{-}06x + 0.2804$.

Bubble Sort: Ascending



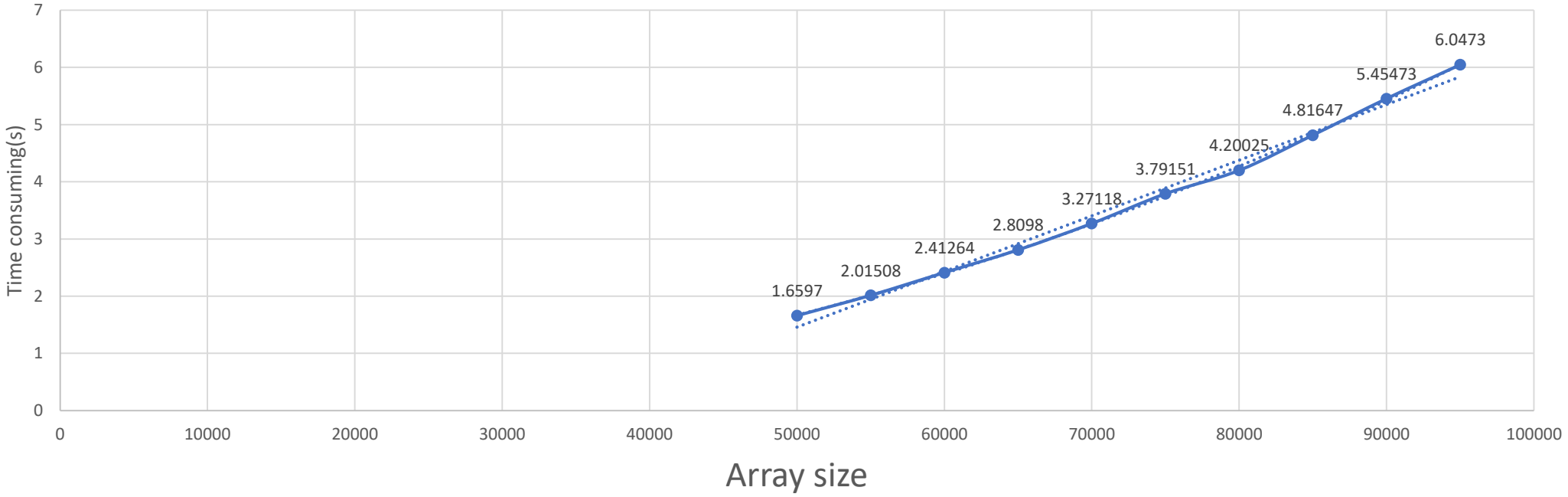
- By my observe, the time consuming is about O(n), because the steep of trending line close to a R^2=0.919 and it is the best case in bubble sort, it should only run one time for each check and do not need to swap.
- I think it will take 0.0384(s), since the trending line is close to a function: y=2E-09x + 0.0184.

Bubble Sort: Descending



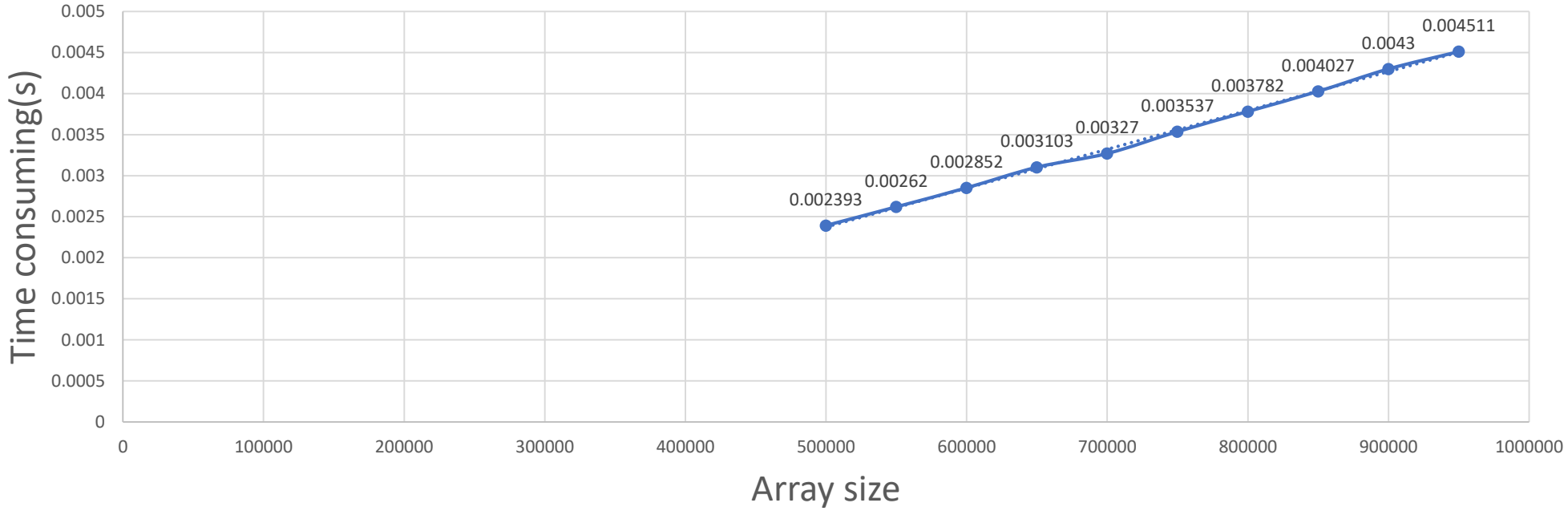
- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=1.
- I think it will take 499800(s), since the trending line is close to a function: $y=5E\text{-}09x^2 - 2E\text{-}05x + 0.4543$.

Insertion Sort: Random



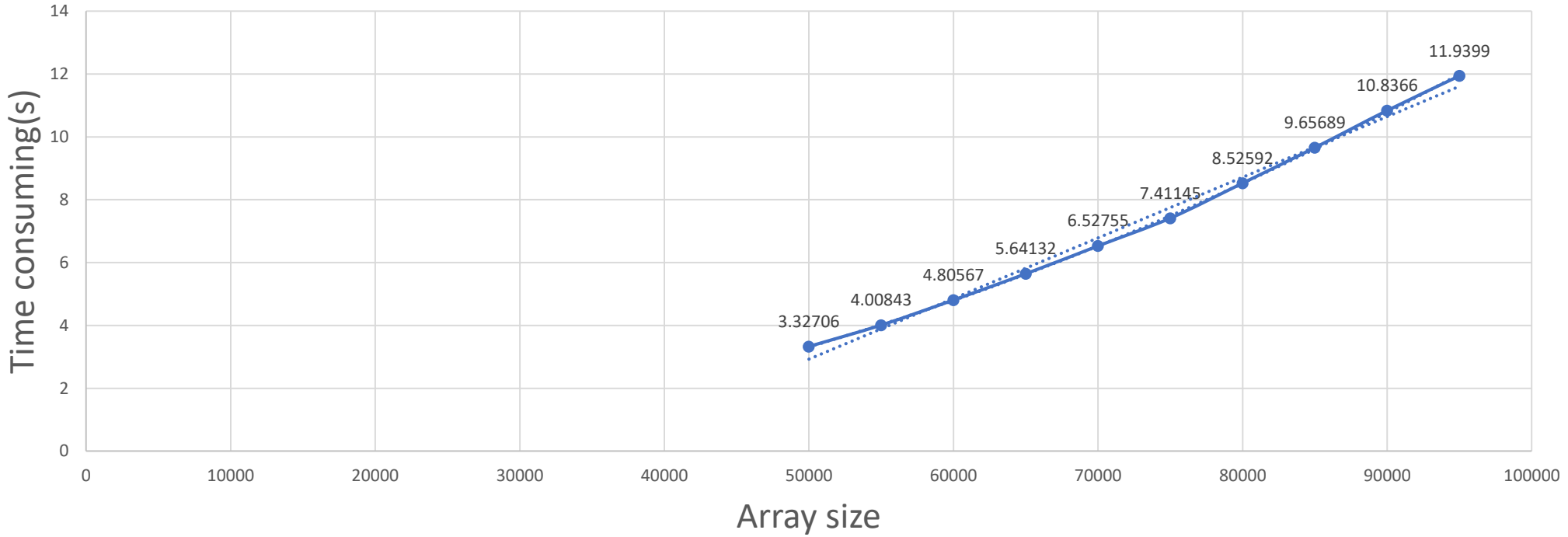
- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=0.9996.
- I think it will take 69950(s), since the trending line is close to a function: $y = 7E\text{-}10x^2 - 5E\text{-}06x + 0.1677$.

Insertion Sort: Ascending



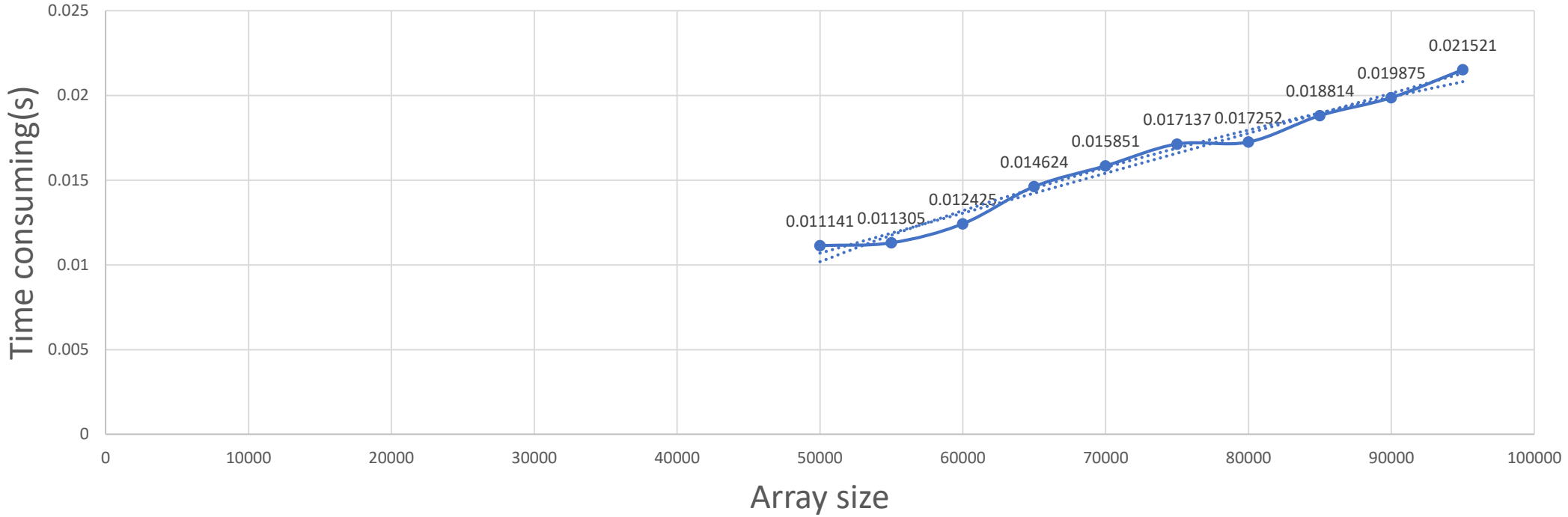
- By my observe, the time consuming is about O(n), because the steep of trending line close to a R^2=0.9989 and this is the best case in Ascending, it need to insert the read number to the right position.
- I think it will take 0.5001(s), since the trending line is close to a function: y=5E-09x + 1E-05.

Insertion Sort: Descending



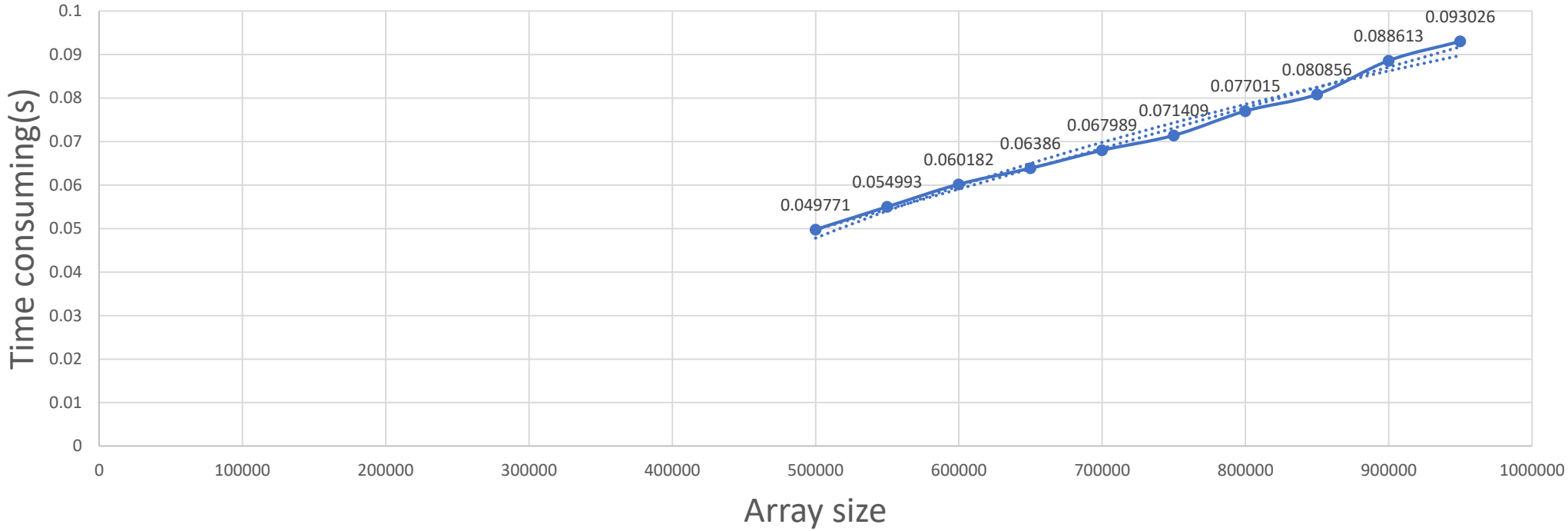
- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=0.9998, it has to swap every time when it reads the next index.
- I think it will take 100050(s), since the trending line is close to a function: $y=1E\text{-}09x^2 + 5E\text{-}06x - 0.1555$.

Merge Sort: Random



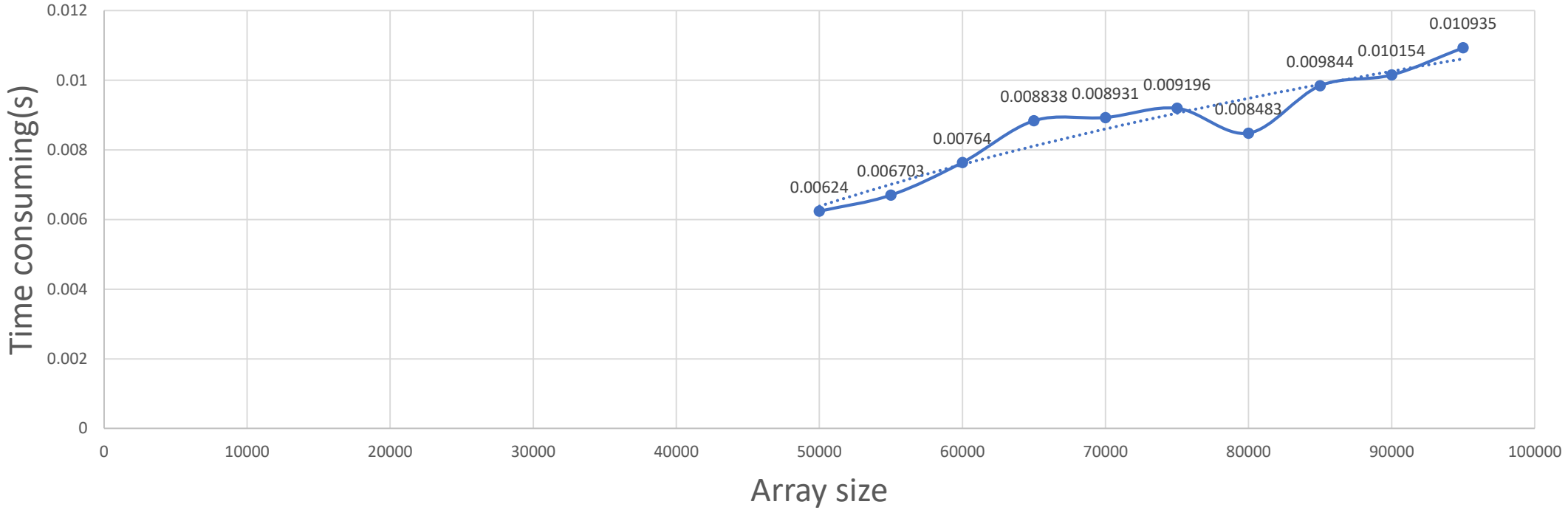
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.9756, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear.
- I think it will take 0.098660388(s), since the trending line is close to a function: y=0.0166ln(x) - 0.1689.

Merge Sort: Ascending



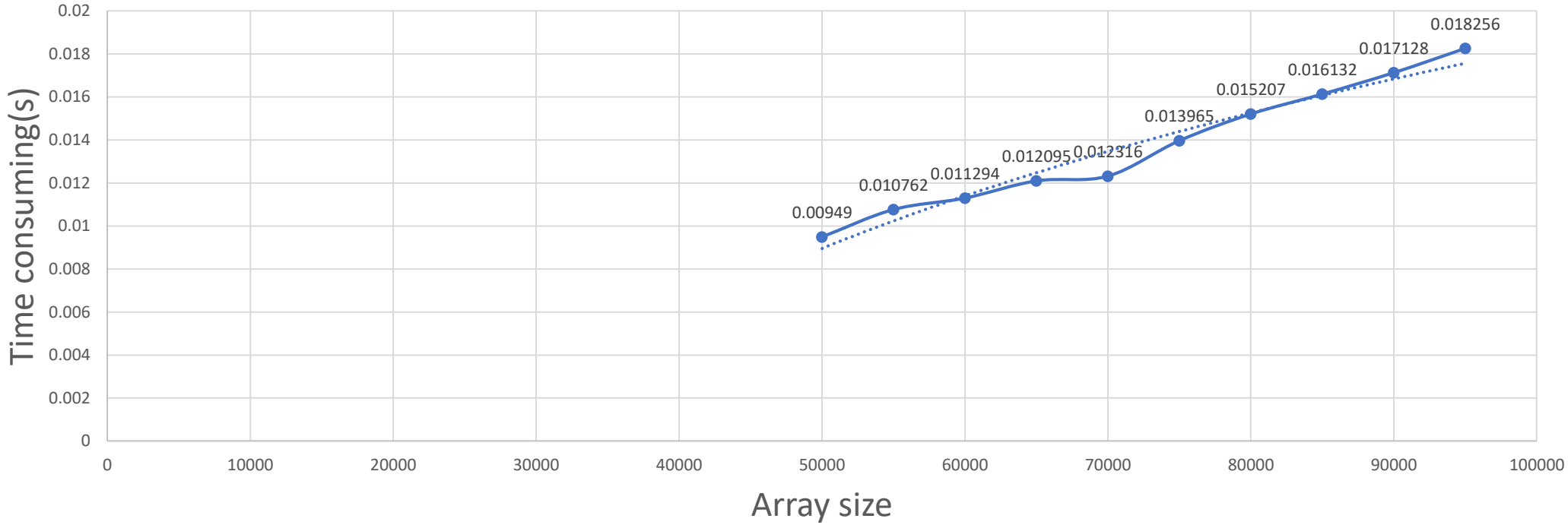
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.9784, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear it did not have best or worst case.
- I think it will take 0.244323456 (s), since the trending line is close to a function: y= 0.0654ln(x) - 0.8098.

Merge Sort: Descending



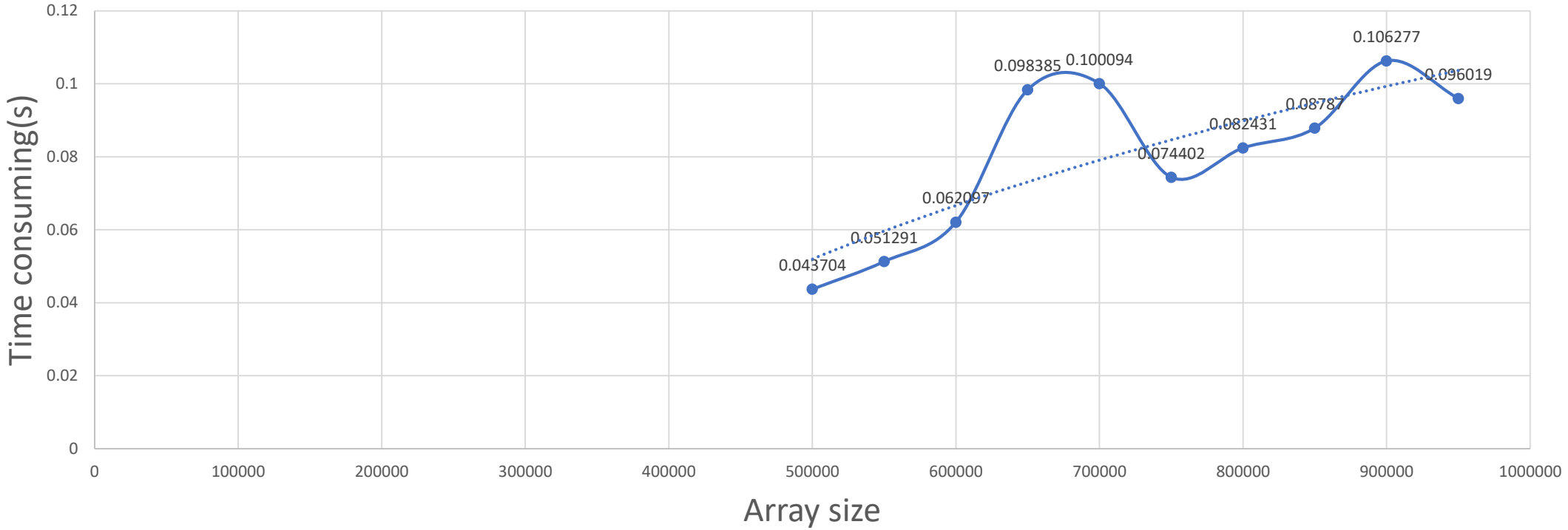
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.9756, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear.
- I think it will take 0.041379431(s), since the trending line is close to a function: y=0.0066ln(x) - 0.065.

Quick Sort: Random



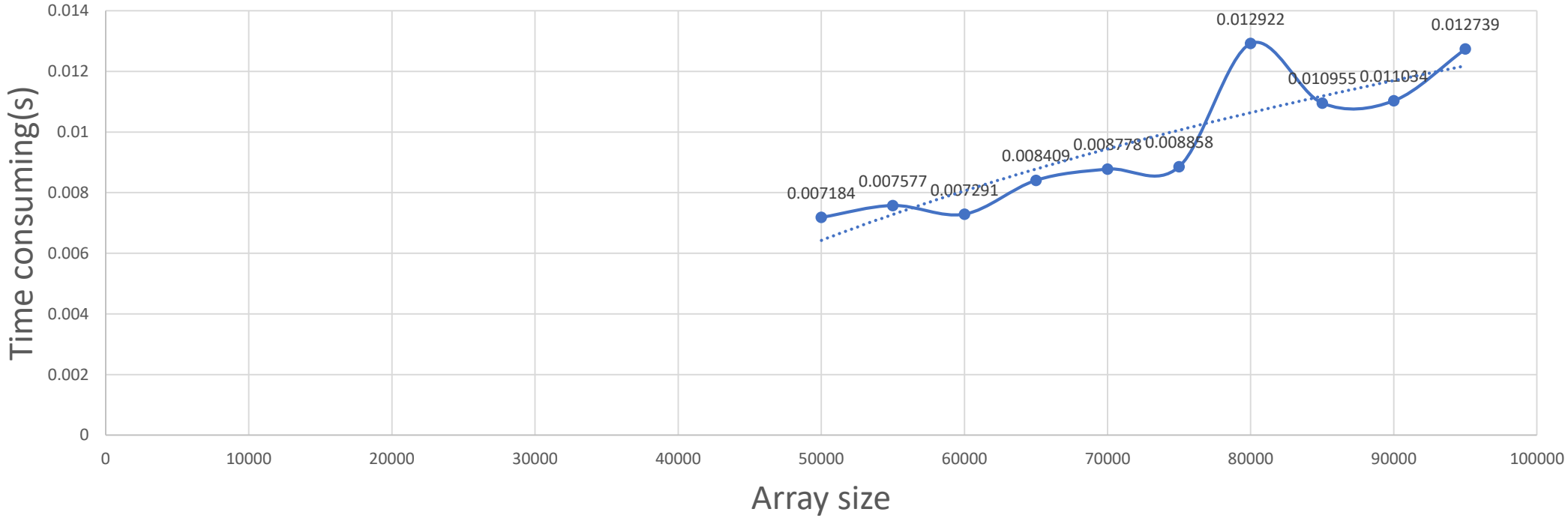
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.9638, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear.
- I think it will take 0.079882482(s), since the trending line is close to a function: y = 0.0134ln(x) - 0.1361.

Quick Sort: Ascending



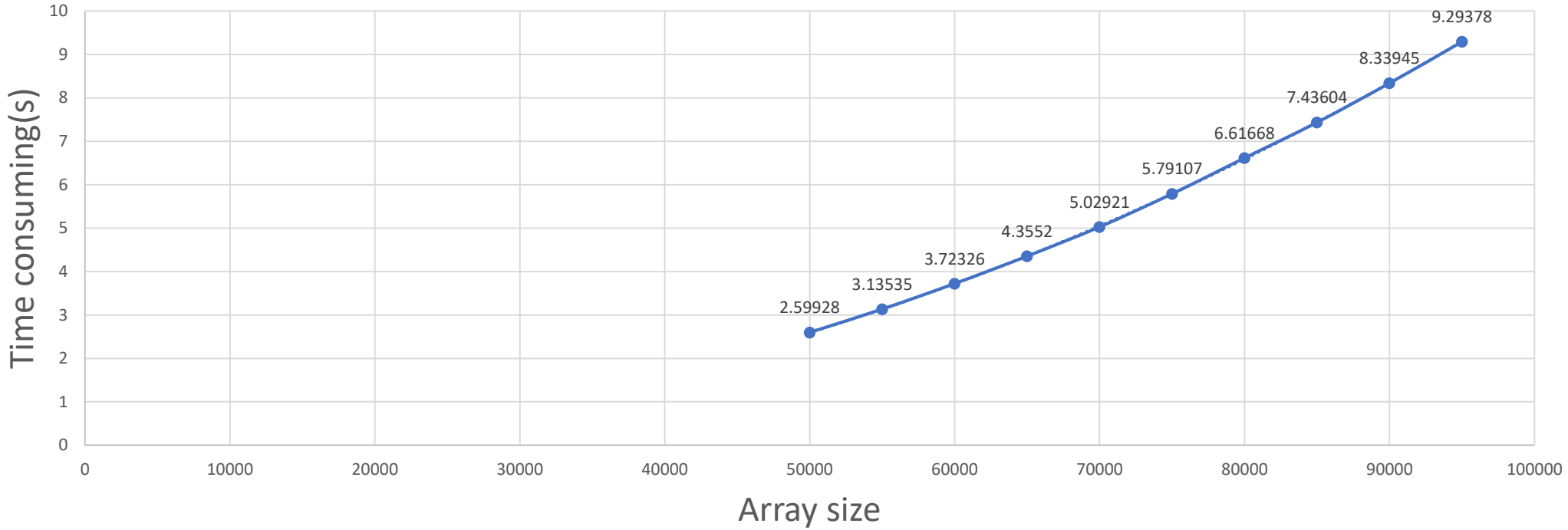
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.6345, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear.
- I think it will take 0.293518509 (s), since the trending line is close to a function: y = 0.0806ln(x) - 1.0056.

Quick Sort: Descending



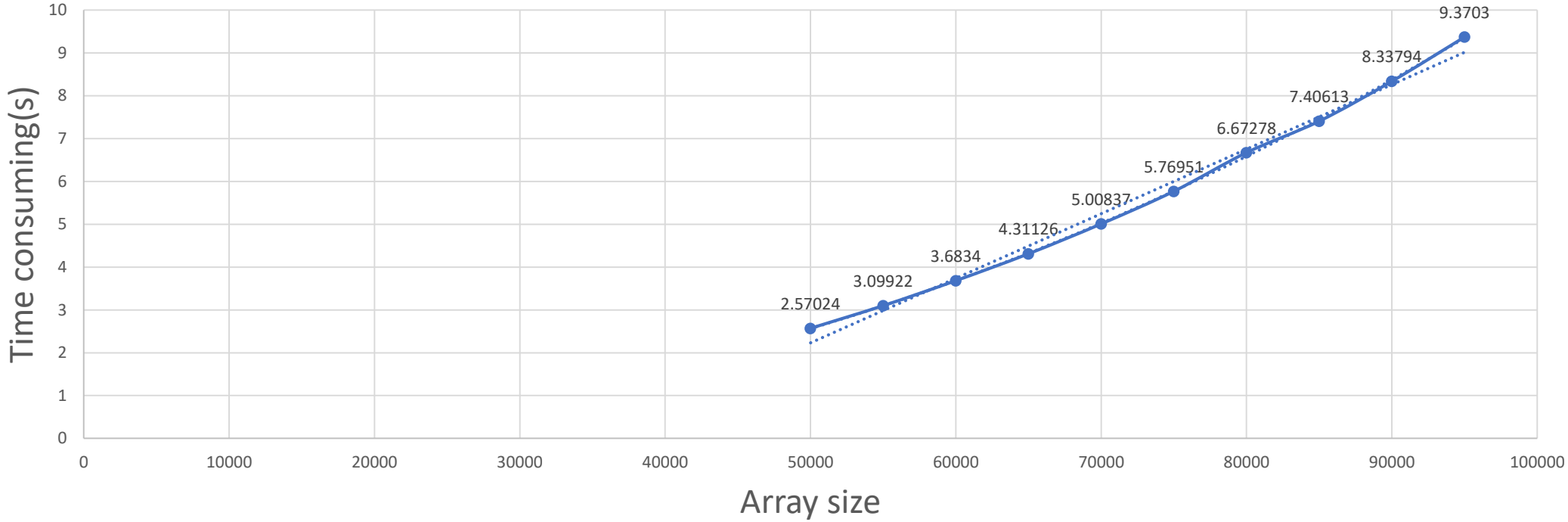
- By my observe, the time consuming is about O(nlog(n)), because the steep of trending line close to a R^2=0.7823, by observing the curve, it is not showing with certain pattern, it is not looks like n^2 and also linear.
- I think it will take 0.05446286 (s), since the trending line is close to a function: y =0.009ln(x) - 0.0906.

Selection Sort: Random



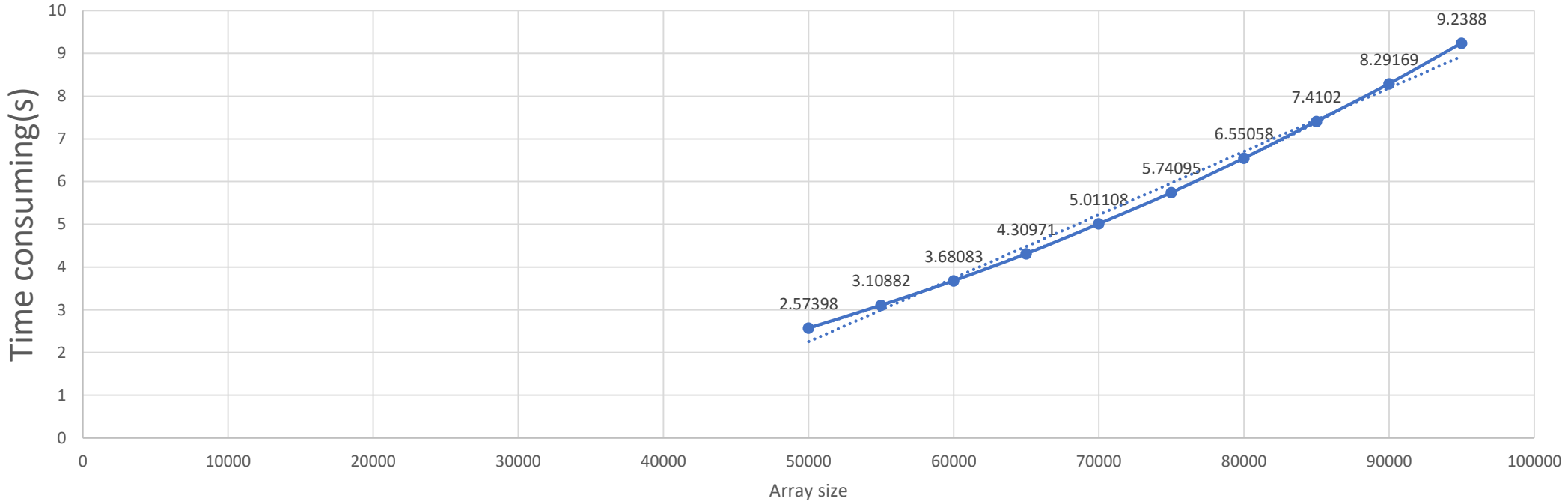
- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=1, since the selection is random and the sequence is not matter, it will go through the array size(N) and find each on to its proper position.
- I think it will take 99960.1606(s), since the trending line is close to a function: $y = 1E\text{-}09x^2 - 4E\text{-}06x + 0.1606$.

Selection Sort: Ascending



- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=0.9998, since the selection is random and the sequence is not matter, it will go through the array size(N) and find each on to its proper position.
- I think it will take 99900.3317(s), since the trending line is close to a function: $y = 1E\text{-}09x^2 - 1E\text{-}05x + 0.3317$.

Selection Sort: Descending



- By my observe, the time consuming is about O(n^2), because the steep of trending line close to a R^2=1, since the selection is random and the sequence is not matter, it will go through the array size(N) and find each on to its proper position.
- I think it will take 99500.1929(s), since the trending line is close to a function: $y = E\text{-}09x^2 - 5E\text{-}06x + 0.1929$.