CS345 Special Assignment

Siddharth Agrawal (150716)
AviRaj (150168)

October,15 2017

Part 3

Brute-Force Algorithm

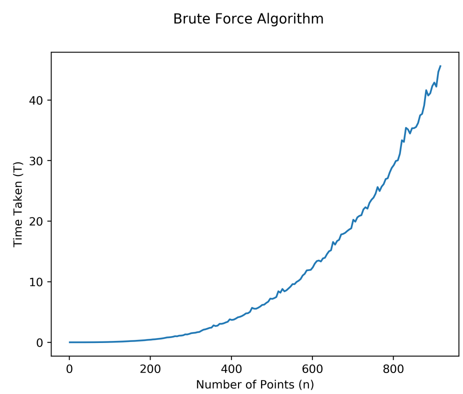

Figure 1: Plot for Running Time of the algorithm for various values of n

Assume, the Running Time (T) is proportional to a polynomial function of n. Hence we check the linearity of graphs - 'T vs n^2', 'T vs n^3' & 'T vs n^4'.

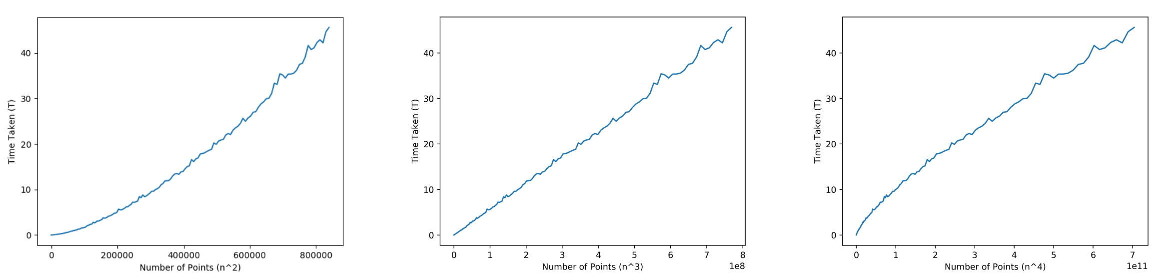

Figure 2: From left to right: 'T vs n^2', 'T vs n^3' & 'T vs n^4'

From the above graphs, we can see that $T \propto n^3$ for n in the range 1 to 1000.
When we tried fitting $O(n^4)$ curve to this distribution, the coefficient of n^4 came out to be $3.025\,364\,92 \times 10^{-11}$.
Thus, this distribution (for n in the range 1 to 1000) is closer to n^3 than n^4.
Theoretically we know that the time bound is $O(n^4)$. But due to hardware constraints, it is not really evident in our graphs.

Randomized Algorithm

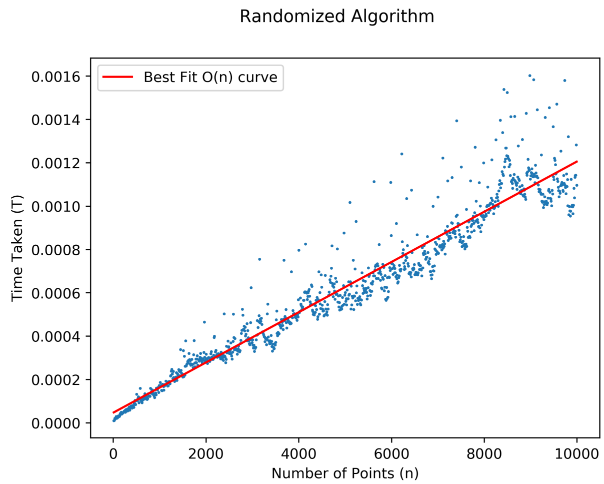

Figure 3: Plot for Running Time of the algorithm for various values of n

From the above graph, it can be clearly seen that $T \propto n$. Thus, time complexity for this algorithm is $O(n)$.

Part 4

In Figure 3, the red curve is the best possible regression curve for the given distribution. Its equation is:

$$T(n) = (4.697\,008\,58 \times 10^{-5}) + (1.159\,335\,27 \times 10^{-7})n$$

The distribution deviates from the regression curve with variance $= 9.483\,645\,598\,090\,948 \times 10^{-5}$.
The empirical distribution of the running time of this algorithm for $n = 10000$ is:

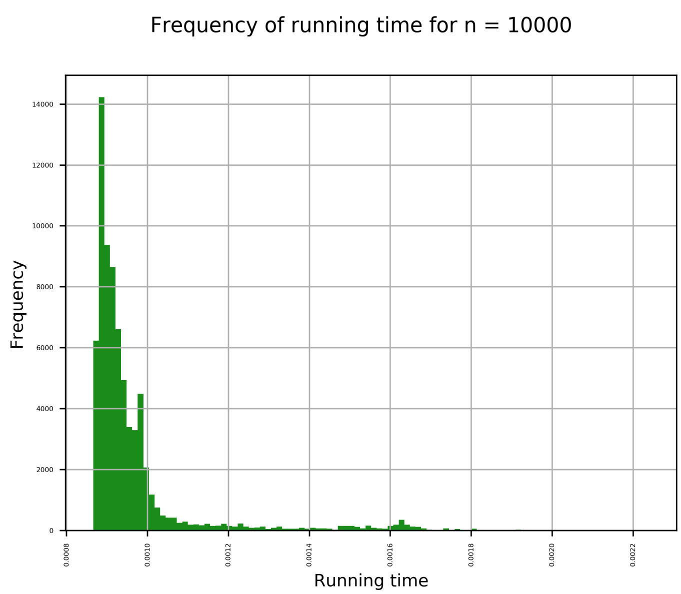

Figure 4: Distribution of Running Time of the algorithm for $n = 10000$