CS325

Miles Curry
Rhea Mae Edwards

PROMGRAMMING ASSIGNMENT 1

PSEUDO-CODE

Brute-Force:

```
Closest-Pair(p_1, p_2, …, p_n) {
      Compute the distances between each pair of the points
      Return the minimum distance and its pair of points
}
```

Naïve Divide and Conquer:

```
Closest-Pair(p_1, p_2, …, p_n) {
if n ≤ 3
      Compute and return the minimum distance     // brute-force method
else
      Pre-Sort all points based on x coordinates
      Compute separation line L
      δ_1 = Closest-Pair (left half)
      δ_2 = Closest-Pair (right half)
      δ = min(δ_1, δ_2)

      Pre-Sort all points based on y coordinates
      Identify all points within δ from L
      Sort them by y-coordinate into M_y
      d_m = closest-cross-pair (M_y, δ)

      return d_m
}
```

Enhanced Divide and Conquer:

```
Closest-Pair(p_1, p_2, …, p_n) {
if n ≤ 3
      Compute and return the minimum distance     // brute-force method
else
      Pre-Sort all points based on x and y coordinates
      Compute separation line L
      δ_1 = Closest-Pair (left half)
      δ_2 = Closest-Pair (right half)
      δ = min(δ_1, δ_2)

      Identify all points within δ from L from pre-sorted list into M_y
      d_m = closest-cross-pair (M_y, δ)

      return d_m
}
```

CS325

Miles Curry
Rhea Mae Edwards

ASYMPTOTIC ANALYSIS OF RUN TIME

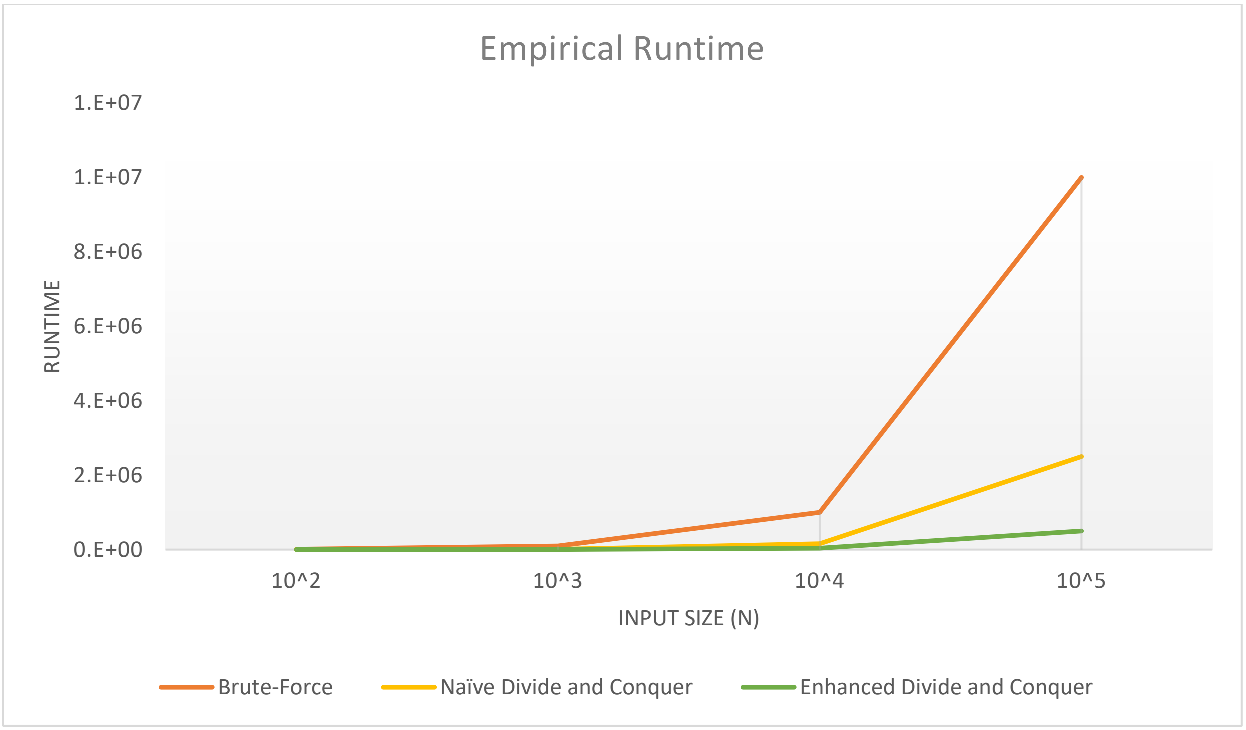
Runtime:

Brute-Force	$O(n^2)$
Naïve Divide and Conquer	$O(n\log^2 n)$
Enhanced Divide and Conquer	$O(n\log n)$

Recursive Relation:

Naïve Divide and Conquer	$T(n) = 2T(n/2) + cn\log n$
Enhanced Divide and Conquer	$T(n) = 2T(n/2) + cn$

PLOTTING THE RUNTIME



INTERPRETATION AND DISCUSSION

The runtime plot illustrates the amount of runtime a certain method of the finding the closest pair of points algorithm based on the various sizes of the inputs put through each of the methods. Each of the methods, brute-force, naïve divide and conquer, and enhanced divide and conquer, appear to be all exponential growth curves on the graph. The brute-force method has the greatest value of runtime being the longest computational method described, followed by the naïve divide and conquer method, and then the enhanced divide and conquer method. The input sizes used were 10^2, 10^3, 10^4, and 10^5, and the values of runtime plotted ranged from the values $2.0\text{x}10^2$ to $1.0\text{x}10^7$.

The growth curves do closely match our expectations based on their theoretical bounds.

CS325

Miles Curry
Rhea Mae Edwards

Some possible explanations for any discrepancy between the experimental runtime and the asymptotic runtime could be the style and structure the code that has been written to achieve the same result and also possibly the type of language the program that has been written in. The way the program is written can affect how the program is compiled and used through the operating system, which can affect the runtime of each of the methods overall. If the program is written in a different language, depending on how the language reads and understands the code, means that it can be computed differently just based on the language the program is written in.