|  |  |  |
| --- | --- | --- |
| N | Brute(in second) | Fast(in second) |
| 5 | 0.031 | 0.031 |
| 10 | 0.032 | 0.029 |
| 20 | 0.032 | 0.034 |
| 40 | 0.036 | 0.035 |
| 80 | 0.053 | 0.039 |
| 160 | 0.242 | 0.044 |
| 320 | 2.711 | 0.079 |
| 500 | 20.558 | 0.224 |
| 640 | 53.667 | 0.340 |
| 1000 | 325.055 | 0.53 |

Youshin Kim

2017-02-21

CS251 Project 2

Brute : (3.27 x 10^-10) \* n(n-1)(n-2)(n-3) + n^**2 ~ n^4 because 4 nested for loops accords input + sorting by brute force takes n^2.**

Fast: ~ (6.70228 × 10^-8)\* n^2+ nlogn ~ n^2 because I have 2 nested for

loops accords input value with sorting by merge sort that takes O(nlogn)

Table of N = {5,10,20,40,80,160,320,500,640,1000}

brute:

fast:

You can see that the plot goes as log sign for fast(concave down) and exponential for brute(concave down).

Estimating N = 1000000 :

Brute (3.27x10^-10)\*(( 1,000,000)(999999)(999998)(999997) + (1000000)^2) = 3.26998×10^14 seconds

Fast (6.70228 × 10^-8)\* (1000000^2+1000000log1000000) = 992977 seconds