Ryan Everett

CS251

Pattern Recognition Analysis

|  |  |  |
| --- | --- | --- |
| **N** | **Brute(N)** | **Fast(N)** |
| **10** | 0.005563s | 0.003563s |
| **20** | 0.012s | 0.007s |
| **40** | 0.018s | 0.011s |
| **50** | 0.024s | 0.013s |
| **80** | 0.1s | 0.021s |
| **100** | 0.177s | 0.041s |
| **150** | 0.776s | 0.043s |
| **200** | 2.272s | 0.067s |
| **250** | 3.639s | 0.135s |
| **300** | 6.931s | 0.074s |
| **350** | 12.926s | 0.192s |
| **400** | 20.036s | 0.185s |
| **1000** | 679.193s | 0.671s |
| **2000** |  | 1.854s |
| **3000** |  | 2.568s |
| **4000** |  | 4.615s |
| **5000** |  | 7.746s |
| **6000** |  | 11.9s |
| **8000** |  | 20.71s |
| **10000** |  | 37.038s |

The difference between the two algorithms is clear from the get-go. The Brute force algorithm begins to slow down very fast; N=400 is the largest tested N value that is sub-200 seconds. Whereas the Fast Algorithm can go all the way to N=10,000 without coming close to the 200 second ceiling. Looking at the Runtime graph of Brute versus Fast, it’s hard to see the difference since Fast stays under one second for the N values. It’s better to plot Fast on it’s own graph to get a proper view of how it grows.

We can use a linear correlation equation for estimating future runtimes. The Brute force algorithm gives us a correlation coefficient of **0.886979** for a sample set of 13 values. The Fast algorithm gives a correlation coefficient of **0.979412** for a sample set of 20 values. These are both good enough for use with predicting n=1,000,000.

The Brute algorithm gives a linear function of **−85.939 + 0.625x** where x=n, so at n=1,000,000 the Brute force algorithm will be around 624,914 s.

The Fast algorithm gives a linear function of **−1.533 + 0.003x** where x=n, so at n=1,000,000 the Brute force algorithm will be around 2,998.47s.