**Casey Levy - CS 325 - HW 1**

**Problem 1**

* < 37 < √N < N < N log(log N) < N log N ≤ N log(N2) < N log2N < N1.5 < N2 < N2 log N < N3 < NN/2 < 2N
* **N log N** and **N log(N2)** are the only functions that grow at the same rate
  + N log(N2) = 2N log N = O(N log N)

**Problem 2**

* claim that N log N < N(1+ )

N(1+ ) = N \* N

N \* N < N log N

N < log N

log N < log log N

Simplified gives us which then shows us that  **< log log N**

**Problem 3**

1. sum = 0; **O(1)**

for( i = 0; i < n; ++i ) **O(n)**

++sum; **O(1)**

**O(1) + O(n) + O(1)**

**Fragment runs in O(n)**

1. sum = 0; **O(1)**

for(i = 0; i < n; ++i) **O(n)**

for(j = 0; j < n; ++j) **O(n)**

++sum; **O(1)**

**O(1) + O(n \* n) + O(1)**

**Fragment runs in O(n2)**

1. sum = 0;  **O(1)**

for(i = 0; i < n; ++i) **O(n)**

for(j = 0; j < n\*n; ++j) **O(n2)**

++sum; **O(1)**

**O(1) + O(n \* n2) + O(1)**

**Fragment runs in O(n3)**

1. sum = 0; **O(1)**

for( i = 0; i < n; ++i ) **O(n)**

for(j = 0; j < i; ++j ) **O(n)**

++sum; **O(1)**

**O(1) + O(n \* n) + O(1)**

**Fragment runs in O(n2)**

1. sum = 0; **O(1)**

for( i = 0; i < n; ++i ) **O(n)**

for( j = 0; j < i \* i; ++j ) **O(n2)**

for( k = 0; k < j; ++k ) **O(n2)**

++sum; **O(1)**

**O(1) + O(n \* n2** \* **n2) + O(1)**

**Fragment runs in O(n5)**

1. sum = 0; **O(1)**

for(i = 1; i < n; ++i) **O(n)**

for(j = 1; j < i\*i; ++j) **O(n2)**

if(j % i == 0) **O(1)**

for(k = 0; k < j; ++k) **O(n2)**

++sum; **O(1)**

**O(1) + O(n \* n2) + O(1) + O(n2) + O(1)**

**Fragment runs in O(n4)**

**Problem 5**

**b and c)** Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

**d) Graphical user interface

Description automatically generated**

**e)**

Based on the theoretical runtime of **Merge Sort**, O (n log(n)), my actual runtimes seem to somewhat follow the trend line of such a function, trending linearly.

Based on the theoretical runtime of **Insert Sort**, O(n2), my actual runtimes seem to grow a bit faster than the theoretical time. Based on the graph above, my runtimes seem to follow more of a trend line of O(2n).