Practical 2:

Exercise 1: Complete these sentences..

1. Algorithms with time complexities such as n and 100n are called linear or linear-time algorithms.
2. Algorithms with time complexities such as n2 are called quadratic-time algorithms (True or False). True
3. Any quadratic-time algorithm is eventually more efficient than any linear-time algorithm (True or False). False
4. Functions such as 5n2 and 5n2 +100 are called quadratic functions.

Exercise 2:

|  |  |
| --- | --- |
| T(N) | Growth Function |
| N^2 | 4 |
| 480 | 1 |
| 2^n | 5 |
| logN | 2 |
| 2^4 | 1 |
| 380N | 3 |
| 1/2N | 3 |

|  |  |
| --- | --- |
| T(N) | Growth Function |
| N logN | 3 |
| N^4 | 5 |
| 2^n | 6 |
| Log8N | 2 |
| nlog4N | 3 |
| Log2N | 2 |
| nlog6N | 3 |
| 300 | 1 |
| 6N^3 | 4 |

|  |  |
| --- | --- |
| T(N) | growth best characterizes each of the functions below? |
| 1 | Constant |
| 2n^3 | Polynomial |
| (4/3)n | Linear |
| 2^n | Exponential |
| 4n^2 | Polynomial |
| 5600 | Constant |
| 2493n | Linear |
| 3/2^n | Exponential |

Exercise 3 :

**Try these ones yourself:**

1. f( n ) = 5n + 12
2. f( n ) = 109
3. f( n ) = n2+ 3n + 112
4. f( n ) = n3 + 1999n + 1337
5. Drop lower order terms, drop the constant factors, keep the terms that grow the fastest, use the smallest possible class of functions
6. Drop the multiplier 109\*1, but leave a 1 as it is a non zero constant. F(n) = 1
7. N^2 grows larger than 3n, so we keep that and f(n) = n^2.
8. N^3 is the largest, so we keep that and drop every other term. F(n) = n^3.

Exercise 4 :

## What is the complexity of the functions below?

1. Linearithmic
2. O(1) constant time
3. O(n)
4. O(n^2) quadradic time
5. O(n)
6. O(1)
7. O(n^2)
8. Big O (logN)
9. O(N^3)

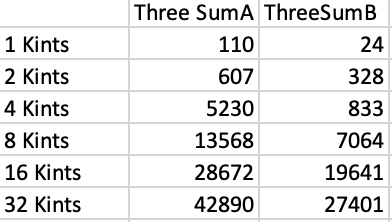
**Exercise 5 :**

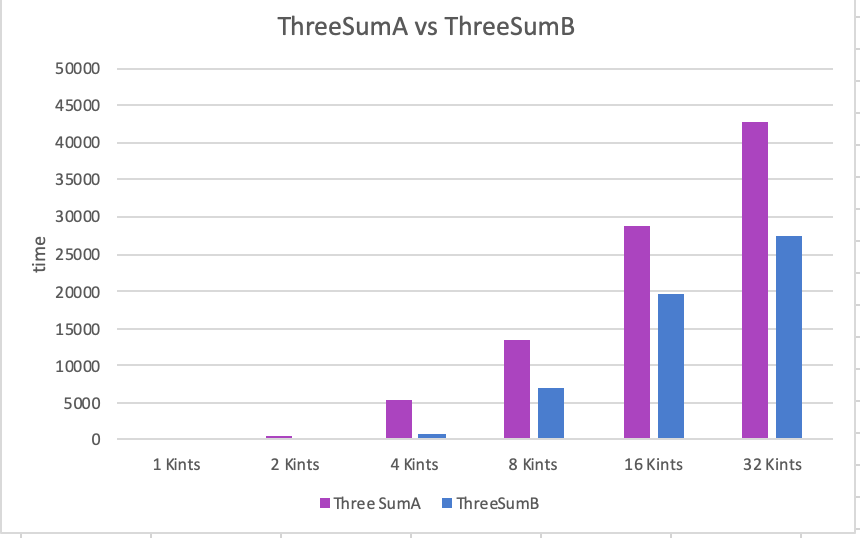
It is easy to see that a possible choice is *c* = 9 and *n*0 = 5. Indeed, this is one of infinitely many choices available because there is a trade-off between *c* and *n*0. For example, we could rely on constants *c* = 13 and *n*0 = 1.

Comparing the two algorithms :

ThreeSumA is N^3

ThreeSumB is N^2 log N





ThreeSumA is slower comparing to ThreeSumB as seen in the graph. This is due to it being n^3. The time complexity of this code has a slower rate compared to n^2 log n. As we can see the time increases for both ThreeSumA and ThreeSumB has the size of the text files increase.