CS 423: Analysis of Algorithms HW 2.1 SOLUTIONS

**COMPLETE Exercises # 1, 3, 8 in Section 2.1 EXTRA 4(a), 10**

1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Problem | Size Metric | Basic Operation | Basic Operation Count | Will the count of the Basic Operation Change **for inputs of same size?** |
| Sum of **N** numbers | n = N = (the **number of numbers being added together)** | Addition of two numbers | n-1 | No. Will **always** do n-1 additions. |
| Computing **N!** | n = N=(The number of numbers being multiplied together) | Multiplication of two numbers | n-1 | No. will always do n-1 multiplications (it will never occur do to the nature of the problem because is a factorial ). |
| Largest element in a list of **N** numbers | n = N (the number of numbers being added in the list) | Comparison of two numbers | If is sorted O(1)  Else n -1 comparison | No because I have to traverse the entire list  And check all numbers so is n-1 |
| Euclid’s algorithm:  **Gcd(M,N)** | n = N ( The number of division/mod in the Z realm that you need to bring the reminder to 0) | Division/mod | The number of division/mod in the Z realm needed to reach mod 0 | Yes because for multiples of the divisor there will be just on operation |
| Sieve of Eratosthenes: find all primes **<= N** | n = N ( With N being (n log log n) | Multiples of up to N | (n log log n) | No it will be always (n log log n) |
| Pen-and-pencil algorithm for multiplying 2 **N-digit** decimal integers | n = N(Size of each digit) | **Count the** intersection diagonally of each line | I don’t know how I can calculate it | No will always have to count the number of intersections  https://www.youtube.com/watch?v=e-P5RGdjICo |

**3)** Depends. We must have a precondition and that is the list is sorted. If the list is not sorted then there will be a chance that the variation won’t work and the sequential search will be the most efficient if not the only one. If the list is sorted then yes. For instance, binary search has an efficiency of O( log n) while linear search is O(n).

**8)** What happens to the given functions if ***n*** is quadrupled?

d. The function’s value will increase by a **FACTOR of 16 or 16 times**:

\* 16;

(Log2n) = n^2 \* 2^2 = 2n^4 = log4 2n;

Sqr(n) = 4sqr(2);

N = 4n

N^2 = (n^2)\*4

N^3 = (n^3)\*4

2^n = 4\*(2^n)

9)

a) < d) <

b) == e) >

c) < f) <