Gang Hyun Kim
40097242
COMP 352
May 19th, 2019

# Assignment #1

## Question 1

(a)

```
count = 0
for i = 0 to n-1 do {
    sum = 0
    for j = 0 to n-1 do {
        sum = sum + A[0]
        for k = 1 to j do
            sum = sum + A[k]
    }
    if B[i] == sum then count = count + 1
```

for k = 1 to j has a time complexity of $O(n^2)$ since it is a summation and by definition a summation is

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$ which makes the time complexity as such

for j = 0 to n-1 has a time complexity of $O(n)$ since it increments in a linear fashion until it reaches n-1

Therefore, the time complexity of the two for loops are $O(n^2)$

for i = 0 to n-1 has a time complexity of $O(n)$

The program as a whole has a time complexity of $O(n \cdot n^2) = O(n^3)$ due to the outer for loop and the other two inner for loops

(b)

| i | j | $sum_1$ | k | $sum_2$ | count |
|---|---|---|---|---|---|
| 0 | 0 | 1 | - | 1 | 0 |
| | 1 | 2 | 1 | 4 | 0 |
| | 2 | 5 | 1 | 7 | 0 |
| | | | 2 | 12 | 0 |
| | 3 | 13 | 1 | 15 | 0 |
| | | | 2 | 20 | 0 |
| | | | 3 | 29 | 0 |
| B[0] = 2, sum = 29 | | | | | 0 |
| 1 | 0 | 1 | - | 1 | 0 |
| | 1 | 2 | 1 | 4 | 0 |
| | 2 | 5 | 1 | 7 | 0 |
| | | | 2 | 12 | 0 |
| | 3 | 13 | 1 | 15 | 0 |
| | | | 2 | 20 | 0 |
| | | | 3 | 29 | 0 |
| B[1] = 29, sum = 29 | | | | | 1 |
| 2 | 0 | 1 | - | 1 | 1 |
| | 1 | 2 | 1 | 4 | 1 |
| | 2 | 5 | 1 | 7 | 1 |
| | | | 2 | 12 | 1 |
| | 3 | 13 | 1 | 15 | 1 |
| | | | 2 | 20 | 1 |
| | | | 3 | 29 | 1 |
| B[2] = 40, sum = 29 | | | | | 1 |
| 3 | 0 | 1 | - | 1 | 1 |
| | 1 | 2 | 1 | 4 | 1 |
| | 2 | 5 | 1 | 7 | 1 |
| | | | 2 | 12 | 1 |
| | 3 | 13 | 1 | 15 | 1 |
| | | | 2 | 20 | 1 |
| | | | 3 | 29 | 1 |
| B[3] = 57, sum = 29 | | | | | 1 |

FINAL OUTPUT = 1

Question 2

(a) for (int i=0; i<n; i=i+c)

for (int j=1; j<1024; j=j*2)

Sum[i]+= j * Sum[i];

This for loop has a time complexity of O(c) since it runs always until 1023.

This for loop has a time complexity of O(n)

↳ $m = \frac{n}{c} \rightarrow O(n)$

This program has a time complexity of O(n)

(b) for (int i=1; i<n; i=i*2)

for (int j=0; j<i; j=j+2)

Sum[i] += j * sum[i];

m = # of iterations

| | | i |
|---|---|---|
| $1 \cdot 2^0$ | = | 1 |
| $1 \cdot 2^1$ | = | 2 |
| $1 \cdot 2^2$ | = | 4 |
| $1 \cdot 2^3$ | = | 8 |
| $1 \cdot 2^4$ | = | 16 |
| ⋮ | | ⋮ |
| $1 \cdot 2^m$ | = | n |

$\rightarrow m = \log n$

↳ $O(\log n)$

| i | # of j iterations | |
|---|---|---|
| 1 | 1 | |
| 2 | 1 | $2^0$ |
| 4 | 2 | $2^1$ |
| 8 | 4 | $2^2$ |
| 16 | 8 | $2^3$ |
| ⋮ | ⋮ | |
| n | | |

$1 + \sum_{m=0}^{m-2} 2^m = 1 + 2^{m-1} - 1$

$= 2^{m-1}$

$2^{\log n - 1} = \frac{1}{2} n$

$\therefore O(n)$

Therefore, the algorithm has a time complexity of O(n)

(c) for (int i = 1; i < n; i = i*2)
for (int j = 1; j < i; j = j*2)
sum [i] += j * sum[i] ;

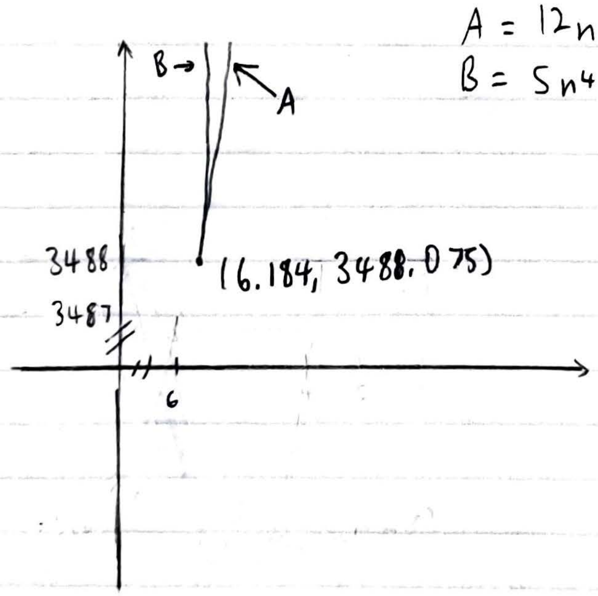
| $i$ | # j is iterated |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 4 | 2 |
| 8 | 4 |
| ⋮ | ⋮ |
| n | |

The time complexity for the inner loop is $O(n^2)$ because to find the total number of iterations, it is a summation which by definition is $\frac{n(n+1)}{2}$ making it $O(n^2)$

The outer loop's time complexity is $O(\log n)$
Therefore, the algorithm's time complexity is $O(\log^2 n)$

Question 3

$A = 12n^3 + 40n\log n$
$B = 5n^4 - 100n^2$



Question 4

(a) If $0 \leq d(n) \leq c_1 \cdot f(n_1)$

$0 \leq e(n) \leq c_2 \cdot g(n_2)$

then $0 \leq d(n) + e(n) \leq c_1 \cdot f(n_1) + c_2 \cdot g(n_2)$

at some point where $c_1 = c_2$

↳ $0 \leq d(n) + e(n) \leq c(f(n_1) + g(n_2))$ where $n > n_1$ if $n_1 > n_2$

$n > n_2$ if $n_2 > n_1$

$n > n_1 = n_2$

$\therefore$ $d(n) + e(n)$ is $O(f(n_1) + g(n_2))$

(b)

Let $2^{n+1}$ be $a(n)$

Let $2^n$ be $b(n)$

Let $a(n)$ is $O(b(n))$

$0 \leq a(n) \leq cb(n)$ if $c \geq 2$

Therefore, the statement $2^{n+1}$ is $O(2^n)$ is true

$2^n > n^3$ for certain numbers such as when $n = 10$

$1024 > 1000$ so $n^3$ is $O(2^n)$.

Therefore, $2^{n+1} + n^3$ is $O(2^n)$

(c)

$$2 \times n! = \overbrace{2 \times 2 \times 3 \ldots \times n}^{n \text{ term}}$$

$$2^n = \underbrace{2 \times 2 \times 2 \ldots \times 2}_{n \text{ terms}}$$

$2^n \leq c \cdot n!$ where $n \geq 0$, $c = 2$

$\therefore$ $2^n$ is $O(n!)$

(d) $\log(n!) = \underbrace{\log(1) + \log(2) + \ldots + \log(n)}_{n \text{ terms}}$

$n\log(n) = \underbrace{\log(n) + \log(n) + \log(n) + \ldots + \log(n)}_{n \text{ terms}}$

$\log(n!) \leq n\log n$ where $n > 0$

$\therefore$ $\log(n!)$ is $O(n\log n)$