Gurumanie Singh Dhiman (COMS3110 HW1)

(1)(a)

$$(3n^2+1)\frac{(4n^3+n)}{2} \in \mathrm{O}(\mathrm{n}^5)$$

$$f(n) = (3n^2+1)\frac{(4n^3+n)}{2}\ , \qquad g(n) = O(n^5)$$

$$f(n) = 3n^2 \cdot \frac{(4n^3+n)}{2} + \frac{(4n^3+n)}{2}$$

$$= \frac{(3n^2)(4n^3+n)}{2} + \frac{(4n^3+n)}{2}$$

$$= \frac{(12n^5+3n^3)}{2} + \frac{(4n^3+n)}{2}$$

$$= \frac{12n^5+3n^3+4n^3+n}{2}$$

$$= \frac{12n^5+7n^3+n}{2}$$

$$= 6n^5 + \frac{7n^3}{2} + \frac{n}{2} \leq 6n^5 + \frac{7n^5}{2} + \frac{n^5}{2}$$

$$= n^5\left(6 + \frac{7}{2} + \frac{1}{2}\right)$$

$$= 10n^5$$

$\therefore$ True, because $\mathrm{f(n)} \in \mathrm{O(g(n))}$: $\exists \mathrm{c} > 0,\ \mathrm{n}_0 > 0$ so that $\forall n \geq n_0\ , f(n) \leq c\big(g(n)\big)$

For our case above, $c = 10, n_0 = 1$

(b) Assume for contradiction that $2^{2^{n+2}} \in O\left(2^{2^{\mathrm{n}+1}}\right)$

Then, $\exists c > 0, n_0 > 0$, such that $2^{2^{n+2}} \leq c \cdot 2^{2^{\mathrm{n}+1}} (\forall n \geq n_0)$

$2^{2^{n+2}} \leq \mathrm{c} \cdot 2^{2^{\mathrm{n}+1}}$

$\frac{2^{2^{n+2}}}{2^{2^{\mathrm{n}+1}}} \leq c$

$\frac{2^{2^{n}.2^{2}}}{2^{2^{n}.2^{1}}} \leq c$

$\frac{2^{2^{n}.4}}{2^{2^{n}.2}} \leq c$

$2^{2^{n}.2} \leq c$

Since $n \rightarrow \infty, 2^{2^{n}.2} \rightarrow \infty, \therefore \forall \mathrm{n}, \nexists \mathrm{c} > 0$

(c) For $a > 1, \; b > 1$,

We need to use the log identity:

$log_a(n) = \frac{log_b(n)}{log_b(a)} \rightarrow log_a(n) = C \cdot log_b(n)$ where $C = \frac{1}{\log_b(a)}$ is a constant

Since Big-O ignores constants,

$O(\log_a(n)) = O(\log_b(n))$

If $f(n) = C \cdot g(n)$ for constant $C > 0$, then $f(n) \in O\big(g(n)\big)$

$\Rightarrow \log_a(n) \in O(\log_b(n)), \quad \log_b(n) \in O(\log_a(n))$

$\therefore$ The two functions are true $\quad \forall a, b > 1, \;\; O(\log_a(n)) = O(\log_b(n))$

(2)(a)

```
a = 0;                                    // c1
for i in the range [1, n]{                // c2
    for j in the range [1, i]{            // c3
        for k in the range [1, i+j]{      // c4
            a++;                          // c5
        }
    }
}
```

$$T(n) = c_1 + \sum_{i=1}^{n} c_2 + \sum_{i=1}^{n}\sum_{j=1}^{i} c_3 + \sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i+j} c_4 + \sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i+j} c_5$$

Replacing c_5 with 1: $T(n) = \sum_{i=1}^{n}\sum_{j=1}^{i}\sum_{k=1}^{i+j} 1 = \sum_{i=1}^{n}\sum_{j=1}^{i}(i+j)$

$$\sum_{j=1}^{i}(i+j) = \sum_{j=1}^{i} i + \sum_{j=1}^{i} j = i^2 + \frac{i(i+1)}{2} = \frac{3i^2+i}{2}$$

$$T(n) = \sum_{i=1}^{n}\frac{3i^2+i}{2} = \frac{3}{2}\sum_{i=1}^{n} i^2 + \frac{1}{2}\sum_{i=1}^{n} i$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \Rightarrow \frac{2n^3+3n^2+n}{6} \in \Theta(n^3), \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \Rightarrow \frac{n^2+n}{2} \in \Theta(n^2)$$

$$\therefore \boldsymbol{T(n) \Rightarrow \Theta(n^3) + \Theta(n^2) \Rightarrow O(n^3)}$$

(b)

```
i = 1;

while i < n {

      for j in range [1, 300] {

            /* constant number of primitive ops */

      }

      i = i * 5

}
```

Outer loop: $i = i \cdot 5 \Rightarrow$ geometric growth

Let $i_k = 5^k$, we stop when $5^k \geq n \Rightarrow k \geq \log_5(n)$

So outer loop runs $\Theta(\log n)$ times

Inner loop: Runs from 1 to 300 (constant time) $\Rightarrow O(1)$ work

$T(n) = \Theta(\log n) \cdot \Theta(1) = O(\log n)$

$$\boldsymbol{T(n) \in O(\log n)}$$

(c) You may assume a naive implementation of pow(a, b) that runs in O(b) time.

```
i = pow(2,n);                       // O(n)

while i ≥ 1 {                       // O(n) + O(n)

    i = i / 2                       // O(n) + O(n) + O(1)

    for j in range [1, 2*i] {       // O(n) + O(n) * O(2^n)

        /* constant number of primitive ops */

    }

}
```

Let $i = 2^n$ and it is stated that pow(2, n) takes $O(n)$ time

Outer loop: $i \to i/2 \Rightarrow$ runs n times since $2^n \to 1$

Inner loop: Each time runs from 1 to $2i$

Total Time: $\sum_{k=0}^{n} O(2^{n-k}) = O(2^n)$

$T(n) = \text{Time to compute} + \text{Time to run all loops} = O(n) + O(2^n) = O(2^n)$

$$\boldsymbol{T(n) \in O(2^n)}$$