Assignment_1: Asymptotic Notations

Name : Omar Bahaa Abd El Moneim Sayed El Ahl

ID : 1200180027

1. We shall evaluate $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ to determine which case is corresponding to each problem, where if limit evalutates to ∞, then $f(n) \in \Omega(g(n))$, if it evaluates to 0, then $f(n) \in \mathrm{O}(g(n))$, and if it evaluates to n where $n \in R$, then $f(n) \in \Theta(g(n))$.

    1.

$$f(n) = 100n + \log n, g(n) = n + (\log n)^2$$
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{100n + \log n}{n + (\log n)^2} = \lim_{n\to\infty} \frac{100 + \frac{1}{n}}{1 + \frac{2*\log n}{n}} = 100$$
$$\because \lim_{n\to\infty} \frac{f(n)}{g(n)} \in R$$
$$\therefore f(n) \in \Theta(g(n))$$

    2.

$$f(n) = \log n, g(n) = \log n^2$$
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\log n}{\log n^2} = \lim_{n\to\infty} \frac{\log n}{2\log n} = \lim_{n\to\infty} \frac{1}{2} = \frac{1}{2}$$
$$\because \lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{1}{2} \in R$$
$$\therefore f(n) \in \Theta(g(n))$$

    3.

$$f(n) = \frac{n^2}{\log n}, g(n) = n(\log n)^2$$
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\frac{n^2}{\log n}}{n(\log n)^2} = \lim_{n\to\infty} \frac{n}{(\log n)^3} = \lim_{n\to\infty} \frac{1}{\frac{3(\log n)^2}{n}} = \lim_{n\to\infty} \frac{n}{\frac{6\log n}{n}} = \lim_{n\to\infty} \frac{n^2}{\frac{6}{n}} = \infty$$
$$\because \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$
$$\therefore f(n) \in \Omega(g(n))$$

    4.

$$f(n) = (\log n)^{\log n}, g(n) = \frac{n}{\log n}$$

Substituting $m = \log n$, we can compare $f_2(m) = m^m$, with $g_2(m) = \frac{2^m}{m}$ as following

$$\lim_{m\to\infty} \frac{f_2(n)}{g_2(n)} = \lim_{m\to\infty} \frac{m^m}{\frac{2^m}{m}} = \lim_{m\to\infty} \frac{m \cdot m^m}{2^m} = \lim_{m\to\infty} m \cdot (\frac{m}{2})^m = \infty$$
$$\because \lim_{m\to\infty} \frac{f_2(m)}{g_2(m)} = \infty$$
$$\therefore f(n) \in \Omega(g(n))$$

5.

$$f(n) = \sqrt{n}, g(n) = (\log n)^5$$
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\sqrt{n}}{(\log n)^5} = \lim_{n\to\infty} \frac{\frac{1}{2\sqrt{n}}}{\frac{5(\log n)^4}{n}} = \lim_{n\to\infty} \frac{n}{10\sqrt{n}(\log n)^4} = \lim_{n\to\infty} \frac{\sqrt{n}}{10(\log n)^4}$$
$$= \lim_{n\to\infty} \frac{n}{2\sqrt{n} \cdot 40(\log n)^3} = \lim_{n\to\infty} \frac{\sqrt{n}}{80(\log n)^3} = \lim_{n\to\infty} \frac{\sqrt{n}}{480(\log n)^2} = \lim_{n\to\infty} \frac{\sqrt{n}}{1920 \log n}$$
$$= \lim_{n\to\infty} \frac{n}{3840\sqrt{n}} = \lim_{n\to\infty} \frac{\sqrt{n}}{3840} = \infty$$
$$\because \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$
$$\therefore f(n) \in \Omega(g(n))$$

6.

$$f(n) = n2^n, g(n) = 3^n$$

Proving that $f(n) \le g(n)$ by induction to decide whether $f(n) \in \mathrm{O}(g(n))$:

- Base case:

  $n_o = 2$, we have $f(2) = 8, g(2) = 9$, we have $f(n) \le g(n)$

- Induction hypothesis:

  Assume $f(k) \le g(k)$, we need to show that

$$f(k) \le g(k) \implies f(k+1) \le g(k+1)$$

Since we have $k2^k \le 3^k$, in order to reach $f(k+1)$ and $g(k+1)$ from $f(k)$ and $g(k)$, respectively, we multiply the former by $\frac{2(k+1)}{k}$ and the latter by 3. And for all $k \ge 2$, where $k \in Z$, we have $\frac{2(k+1)}{k} < 3$.

Thus $f(k) \le g(k) \implies f(k+1) \le g(k+1)$

$f(n) \le g(n)$, thus $f(n) \in \mathrm{O}(g(n))$

7.

$$f(x) = 2^{\sqrt{\log n}}, g(n) = \sqrt{n}$$

$$\because \sqrt{n} \in \Omega(2^{\sqrt{\log n}})$$
$$\therefore f(x) \in \mathrm{O}(g(n))$$

2.

- **Unit cost**:

  Algorithm runs through a loop of $\log m = n$ times, each time we do $\mathrm{O}(1)$, thus unit cost is $\mathrm{O}(n)$
- **Bit cost**

  Algorithm runs through a loop of $\log m = n$ times, each time we do $\mathrm{O}(l^2)$ where l is the number of bits and l^2 is the bit cost of multiplication in an iteration.

  The value of y in each iteration is 2^{2^i} where i is iteration number, thus each iteration require $\mathrm{O}((\log y)^2) = \mathrm{O}((\log 2^{2^i})^2) = \mathrm{O}(2^{2i})$. Summing along the iterations:

$$\sum_{i=1}^{\log m} c2^{2i} = c\sum_{i=1}^{\log m} 4^i = 4c\sum_{i=0}^{\log m-1} 4^i = 4c\frac{4^{\log m}-1}{4-1} \in \mathrm{O}(4^{\log m}) = \mathrm{O}(4^n)$$

3. Given a graph, G, with set of nodes and vertices N and V, respectively.

```
for all v in V: set v.group = 0
function isBipartite(G, s):
   group <- 1
   q <- Queue(s)
   while q:
      u <- q.dequeue()
      for v in u.neighbors():
         if v.group == 0:
            v.group <- group
            q.enqueue(v)
         elif v.group == u.group:
            return False
      group <- group * -1
   return True
```

4.

Expression	**Dominant term(x)**	$\mathrm{O}(\ldots)$
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	$\mathrm{O}(n^3)$
$500n + 100n^{1.5} + 50n\log_{10} n$	$100n^{1.5}$	$\mathrm{O}(n^{1.5})$
$0.3n + 5n^{1.5} + 2.5n^{1.75}$	$2.5n^{1.75}$	$\mathrm{O}(n^{1.75})$
$n^2\log_2 n + n(\log_2 n)^2$	$n^2\log_2 n$	$\mathrm{O}(n^2\log n)$
$n\log_3 n + n\log_2 n$	$n\log_2 n$	$\mathrm{O}(n\log n)$
$3\log_8 n + \log_2\log_2\log_2 n$	$3\log_8 n$	$\mathrm{O}(\log n)$

Expression	Dominant term(x)	$O(\ldots)$
$100n + 0.01n^2$	$0.01n^2$	$O(n^2)$
$0.01n + 100n^2$	$100n^2$	$O(n^2)$
$2n + n^{0.5} + 0.5n^{1.25}$	$0.5n^{1.25}$	$O(n^{1.25})$
$0.01n \log_2 n + n(\log_2 n)^2$	$n(\log_2 n)^2$	$O(n(\log n)^2)$
$100n \log_3 n + n^3 + 100n$	n^3	$O(n^3)$
$0.003 \log_4 n + \log_2 \log_2 n$	$0.003 \log_4 n$	$O(\log n)$

5.

Statement	True/False	correct formula, if false
Rule of sums: $O(f+g) = O(f) + O(g)$	False	$O(f+g) = O(\max(f,g))$
Rule of products: $O(f \cdot g) = O(f) \cdot O(g)$	True	
Transitivity: if $g = O(f)$ and $h = O(f)$ then $g = O(h)$	False	if $g = O(f)$ and $f = O(h)$ then $g = O(h)$
$5n + 8n^2 + 100n^3 = O(n^4)$	True	
$5n + 8n^2 + 100n^3 = O(n^2 \log n)$	False	$5n + 8n^2 + 100n^3 = O(n^3)$

6.

1. $O(f_4) > O(f_1) = O(f_2) = O(f_3)$
2. $O(f_2) > O(f_3) > O(f_4) > O(f_1)$
3. $O(f_2) > O(f_3) > O(f_1) > O(f_4)$