*Note*: Your TA probably will not cover all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. They are deliberately made long so they can serve as a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

In this class, we care a lot about the runtime of algorithms. However, we don't care too much about concrete performance on small input sizes (most algorithms do well on small inputs). Instead we want to compare the *asymptotic (i.e. long-term)* growth of the runtimes.

**Asymptotic Notation:** The following are definitions for $\mathcal{O}(\cdot), \Theta(\cdot)$, and $\Omega(\cdot)$:

- $f(n) = \mathcal{O}(g(n))$ if there exists a $c > 0$ where after large enough $n$, $f(n) \leq c \cdot g(n)$. *(Asymptotically, $f$ grows at most as much as $g$)*
- $f(n) = \Omega(g(n))$ if $g(n) = \mathcal{O}(f(n))$. *(Asymptotically, $f$ grows at least as much as $g$)*
- $f(n) = \Theta(g(n))$ if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$. *(Asymptotically, $f$ and $g$ grow at the same rate)*

If we compare these definitions to the order on the numbers, $\mathcal{O}$ is a lot like $\leq$, $\Omega$ is a lot like $\geq$, and $\Theta$ is a lot like $=$ (except all are with regard to asymptotic behavior).

## 1 Asymptotics and Limits

If we would like to prove asymptotic relations instead of just using them, we can use limits.

**Asymptotic Limit Rules:** If $f(n), g(n) \geq 0$:

- If $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$, then $f(n) = \mathcal{O}(g(n))$.
- If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$, for some $c > 0$, then $f(n) = \Theta(g(n))$.
- If $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$, then $f(n) = \Omega(g(n))$.

Note that these are all sufficient (and not necessary) conditions involving limits, and are not true definitions of $\mathcal{O}$, $\Theta$, and $\Omega$. We highly recommend checking on your own that these statements are correct!)

(a) Prove that $n^3 = \mathcal{O}(n^4)$.

**Solution:**

$$\lim_{n\to\infty} \frac{n^3}{n^4} = \lim_{n\to\infty} \frac{1}{n} = 0$$

So $f(n) = \mathcal{O}(g(n))$

(b) Find an $f(n), g(n) \geq 0$ such that $f(n) = \mathcal{O}(g(n))$, yet $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$.

**Solution:** Let $f(n) = 3n$ and $g(n) = 5n$. Then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{3}{5}$, meaning that $f(n) = \Theta(g(n))$. However, it's still true in this case that $f(n) = \mathcal{O}(g(n))$ (just by the definition of $\Theta$).

(c) Prove that for any $c > 0$, we have $\log n = \mathcal{O}(n^c)$.

*Hint:* Use L'Hôpital's rule: If $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$, then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$ (if the RHS exists)

**Solution:** By L'Hôpital's rule,

$$\lim_{n\to\infty} \frac{\log n}{n^c} = \lim_{n\to\infty} \frac{n^{-1}}{cn^{c-1}} = \lim_{n\to\infty} \frac{1}{cn^c} = 0$$

Therefore, $\log n = \mathcal{O}(n^c)$.

(d) Find an $f(n), g(n) \geq 0$ such that $f(n) = \mathcal{O}(g(n))$, yet $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ does not exist. In this case, you would be unable to use limits to prove $f(n) = \mathcal{O}(g(n))$.

*Hint: think about oscillating functions!*

**Solution:** Let $f(x) = x(\sin x + 1)$ and $g(x) = x$. As $\sin x + 1 \leq 2$, we have that $f(x) \leq 2 \cdot g(x)$ for $x \geq 0$, so $f(x) = \mathcal{O}(g(x))$.

However, if we attempt to evaluate the limit, $\lim_{x\to\infty} \frac{x(\sin x+1)}{x} = \lim_{x\to\infty} \sin x + 1$, which does not exist (sin oscillates forever).

## 2 Asymptotic Complexity Comparisons

(a) Order the following functions so that for all $i, j$, if $f_i$ comes before $f_j$ in the order then $f_i = O(f_j)$. Do not justify your answers.

- $f_1(n) = 3^n$
- $f_2(n) = n^{\frac{1}{3}}$
- $f_3(n) = 12$
- $f_4(n) = 2^{\log_2 n}$
- $f_5(n) = \sqrt{n}$
- $f_6(n) = 2^n$
- $f_7(n) = \log_2 n$
- $f_8(n) = 2^{\sqrt{n}}$
- $f_9(n) = n^3$

As an answer you may just write the functions as a list, e.g. $f_8, f_9, f_1, \ldots$

**Solution:** $f_3, f_7, f_2, f_5, f_4, f_9, f_8, f_6, f_1$

(b) In each of the following, indicate whether $f = O(g)$, $f = \Omega(g)$, or both (in which case $f = \Theta(g)$). ***Briefly*** justify each of your answers. Recall that in terms of asymptotic growth rate, constant < logarithmic < polynomial < exponential.

| | $f(n)$ | $g(n)$ |
|---|---|---|
| (i) | $\log_3 n$ | $\log_4(n)$ |
| (ii) | $n\log(n^4)$ | $n^2\log(n^3)$ |
| (iii) | $\sqrt{n}$ | $(\log n)^3$ |
| (iv) | $n + \log n$ | $n + (\log n)^2$ |

**Solution:**

(i) $f = \Theta(g)$; using the log change of base formula, $\frac{\log n}{\log 3}$ and $\frac{\log n}{\log 4}$ differ only by a constant factor.

(ii) $f = O(g)$; $f(n) = 4n\log(n)$ and $g(n) = 3n^2\log(n)$, and the polynomial in $g$ has the higher degree.

(iii) $f = \Omega(g)$; any polynomial dominates a product of logs. We can also obtain this result via the limit proof below:

$$\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} &= \lim_{n\to\infty} \frac{\sqrt{n}}{(\log n)^3} \\
&= \lim_{n\to\infty} \frac{\frac{1}{2\sqrt{n}}}{3(\log n)^2 \cdot \frac{1}{n}} && \text{[L'Hôpital's rule]} \\
&= \lim_{n\to\infty} \frac{\sqrt{n}}{6(\log n)^2} \\
&= \lim_{n\to\infty} \frac{\sqrt{n}}{24 \log n} && \text{[L'Hôpital's rule again]} \\
&= \lim_{n\to\infty} \frac{\sqrt{n}}{48} && \text{[L'Hôpital's rule one more time]} \\
&= \infty
\end{aligned}$$

(iv) $f = \Theta(g)$; Both $f$ and $g$ grow as $\Theta(n)$ because the linear term dominates the other. We can also obtain this result via the limit proof below:

$$\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} &= \lim_{n\to\infty} \frac{n + \log n}{n + (\log n)^2} \\
&= \lim_{n\to\infty} \frac{1 + \frac{1}{n}}{1 + \frac{2\log n}{n}} && \text{[L'Hôpital's rule]} \\
&= 1
\end{aligned}$$

## 3 Recurrence Relations

**Master Theorem:** If the recurrence relation is of the form $T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$ for some constants $a > 0$, $b > 1$, and $d \geq 0$, then

$$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

Remember that if the recurrence relation is *not* in the form $T(n) = aT\left(\frac{n}{b}\right) + O(n^d)$, you can't use Master Theorem! If this is the case, you can try the following strategies:

1. **"Unraveling" the recurrence relation.**

   We try to recursively keep on plugging in the smaller subproblems (e.g. $T(n/2)$ or $T(n-1)$) to $T(n)$ to find a pattern or simply directly compute the entire expression.

2. **Draw a tree!**

   We use a tree representation to count the total number of calls on each subproblem, doing so by summing up the work per level.

3. **Change of Variables**

   We can make a change of variables (kind of like $u$-substitution for integration) to mold our recurrence into a nicer form to work with.

4. **Squeeze**

   For certain recurrence relations where we can't directly compute the solution, we can indirectly solve it by computing upper and lower bounds for the runtime based on known recurrence relations. In particular, if the upper and lower bounds have the same asymptotic growth, then we have our answer!

5. **Squeeze + Guess & Check**

   If we try using the previous method and can't end up with asymptotically equivalent upper/lower bounds, then we resort to guess-and-checking reasonable runtimes between the bounds to arrive at the solution (look up "The Substitution Method for Solving Recurrences – Brilliant" to see how to do this). For the purposes of this class, if you ever have to resort to using this method, the expression for $T(n)$ will always look "nice."

There are a lot more strategies that are out-of-scope for this class, and if you're curious we highly recommend you to read about them in the following link: `http://beta.iiitdm.ac.in/Faculty_Teaching/Sadagopan/pdf/DAA/new/recurrence-relations-V3.pdf`.

Solve the following recurrence relations, assuming base cases $T(0) = T(1) = 1$:

(a) $T(n) = 2 \cdot T(n/2) + O(n)$

**Solution:** We can use the Master Theorem here! Noting that $a = 2, b = 2$, and $d = 1$, we have that $\log_b a = \log_2(2) = 1 = d$. Thus, via the Master Theorem, we have

$$T(n) = O(n^d \log n) = O(n \log n)$$

(b) $T(n) = T(n-1) + n$

**Solution:** Since we can't use Master Theorem here, we use the "unravelling" strategy as follows:

$$\begin{aligned}
T(n) &= T(n-1) + n \\
&= (T(n-2) + (n-1)) + n \\
&= ((T(n-3) + (n-2)) + (n-1)) + n \\
&= \cdots \text{Unravelling} \cdots \\
&= T(1) + 2 + 3 + \cdots (n-2) + (n-1) + n \\
&= 1 + 2 + 3 + \cdots + (n-2) + (n-1) + n \\
&= \sum_{i=1}^{n} i \\
&= \frac{n(n+1)}{2} \\
&= O(n^2)
\end{aligned}$$

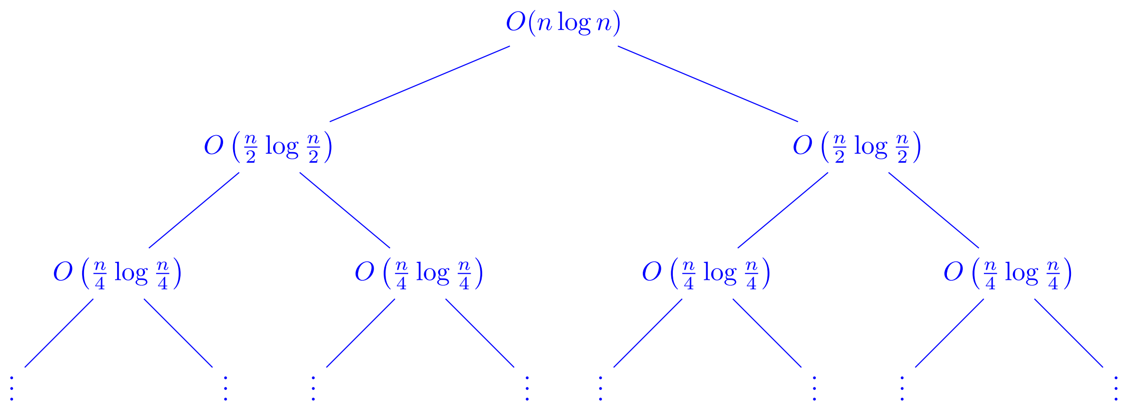
(c) $T(n) = 3 \cdot T(n-2) + 5$

**Solution:** More unravelling! Here we go again:

$$\begin{aligned}
T(n) &= 3T(n-2) + 5 \\
&= 3^2 T(n-4) + 5 \cdot 3 + 5 \\
&= 3^3 T(n-6) + 5 \cdot 3^2 + 5 \cdot 3 + 5 \\
&= 3^4 T(n-8) + 5 \cdot 3^3 + 5 \cdot 3^2 + 5 \cdot 3 + 5 \\
&= \cdots \\
&= 3^{\lfloor n/2 \rfloor} T(n \bmod 2) + 5 \cdot 3^{\lfloor n/2 \rfloor - 1} + 5 \cdot 3^{\lfloor n/2 \rfloor - 2} + \cdots + 5 \cdot 3^2 + 5 \cdot 3 + 5 \\
&= 1 \cdot 3^{\lfloor n/2 \rfloor} + 5 \cdot 3^{\lfloor n/2 \rfloor - 1} + 5 \cdot 3^{\lfloor n/2 \rfloor - 2} + \cdots + 5 \cdot 3^2 + 5 \cdot 3 + 5 \\
&= 3^{\lfloor n/2 \rfloor} + \frac{5 \cdot (3^{\lfloor n/2 \rfloor} - 1)}{3 - 1} \\
&= \frac{7}{2} \cdot 3^{\lfloor n/2 \rfloor} - \frac{5}{2} \\
&= O(3^{n/2})
\end{aligned}$$

(d) $T(n) = 2 \cdot T(n/2) + O(n \log n)$

**Solution:** We use the tree drawing technique here. We draw the following recursion tree, where the nodes represent the work done by a recursive call:


$O(n \log n)$
$O\left(\frac{n}{2} \log \frac{n}{2}\right)$
$O\left(\frac{n}{2} \log \frac{n}{2}\right)$
$O\left(\frac{n}{4} \log \frac{n}{4}\right)$
$O\left(\frac{n}{4} \log \frac{n}{4}\right)$
$O\left(\frac{n}{4} \log \frac{n}{4}\right)$
$O\left(\frac{n}{4} \log \frac{n}{4}\right)$

Summing up all the levels, we get

$$\begin{aligned}
T(n) &= O(n \log n) + O\left(n \log\left(\frac{n}{2}\right)\right) + O\left(n \log\left(\frac{n}{4}\right)\right) + \cdots + O(1) \\
&= O\left(\sum_{i=0}^{\lfloor \log n \rfloor} n \log\left(\frac{n}{2^i}\right)\right) \\
&= O\left(\sum_{i=0}^{\lfloor \log n \rfloor} n \left(\log n - i\right)\right) \\
&= O\left(n\left(\log^2 n - \sum_{i=0}^{\lfloor \log n \rfloor} i\right)\right) \\
&= O\left(n\left(\log^2 n - \frac{1}{2}\log^2 n\right)\right) \\
&= O(n \log^2 n)
\end{aligned}$$

(e) $T(n) = 3T(n^{1/3}) + O(\log n)$

**Solution:** Since we're recursing on a weird $n^{1/3}$ cubic root, we should use a *change of variables* to get mold the recurrence into something that's more manageable. Let's try the substitution $x = \log n$, so that

$$n^{1/3} = (e^x)^{1/3} = e^{x/3}.$$

Then, our recurrence becomes $T(e^x) = 3T(e^{x/3}) + O(x)$. Now, let us define $S(x) = T(e^x) = T(n)$, which has the following (nice) recurrence:

$$S(x) = 3S(x/3) + O(x).$$

Look at this; we can apply the Master Theorem! Solving, we get $S(x) = O(x \log x)$. Finally, note that we want to find $T(n)$ and not $S(x)$, so we plug $x = \log n$ back in as follows:

$$T(n) = S(x) = O(x \log x) = O(\log n \cdot \log \log n).$$

(f) $T(n) = T(n-1) + T(n-2)$

**Solution:** We apply the squeeze + guess & check method. First, we can lower bound it by, $T(n) \geq 2T(n-2)$, so we know $T(n) = \Omega(2^{n/2})$. We can also upper-bound it by $T(n) \leq 2T(n-1)$, which gives $T(n) = O(2^n)$.

Hence, $T(n) = 2^{\Theta(n)}$. However, we can actually compute a more precise runtime! Since we know the runtime is exponential with respect to $n$, we can write the runtime in the form $T(n) = \Theta(a^n)$. Then, plugging this into the recurrence, we have

$$a^n = a^{n-1} + a^{n-2}$$

$$a^2 = a + 1$$
$$a^2 - a - 1 = 0$$

where we can divide by $a^{n-2}$ since $a \neq 0$. By the quadratic formula, we get that $a = \frac{1 \pm \sqrt{5}}{2}$. Since $a$ must be positive, we conclude that $a = \frac{1+\sqrt{5}}{2}$ and thus

$$T(n) = \Theta\left(\left(\frac{1+\sqrt{5}}{2}\right)^n\right)$$

## 4 Complex Numbers Review

A *complex number* is a number that can be written in the rectangular form $a + bi$ ($i$ is the imaginary unit, with $i^2 = -1$). The following famous equation (*Euler's formula*) relates the polar form of complex numbers to the rectangular form:

$$re^{i\theta} = r(\cos\theta + i\sin\theta)$$

where $a = r\cos\theta$ and $b = r\sin\theta$. In polar form, $r \geq 0$ represents the distance of the complex number from 0, and $\theta$ represents its angle. Note that since $\sin(\theta) = \sin(\theta + 2\pi), \cos(\theta) = \cos(\theta + 2\pi)$, we have $re^{i\theta} = re^{i(\theta+2\pi)}$ for any $r, \theta$.

The $n$-th *roots of unity* are the $n$ solutions to the equation $\omega^n = 1$. They are given by

$$\omega_k = e^{2\pi ik/n}, \qquad k = 0, 1, 2, \ldots, n-1$$

(a) Let $x = e^{2\pi i3/10}, y = e^{2\pi i5/10}$ which are two 10-th roots of unity. Compute the product $z = x \cdot y$. Is this an $n$-th root of unity for some $n$? Is it a 10-th root of unity?

What happens if $x = e^{2\pi i6/10}, y = e^{2\pi i7/10}$?

*For all your answers, simplify if possible.*

**Solution:** $z = x \cdot y = e^{2\pi i8/10}$. This is always an 10-th root of unity (it is in general). But because $8/10 = 4/5$, this is also a 5th root of unity.

If $x = e^{2\pi i6/10}, y = e^{2\pi i7/10}$, then we 'wind around' and the product becomes $z = e^{2\pi i13/10} = e^{2\pi i3/10}$.

(b) Show that for any $n$-th root of unity $\omega \neq 1$, $\sum_{k=0}^{n-1} \omega^k = 0$, when $n > 1$.

*Hint: Use the formula for the sum of a geometric series* $\sum_{k=0}^{n} \alpha^k = \frac{\alpha^{n+1}-1}{\alpha-1}$. *It works for complex numbers too!*

**Solution:** Remember that $\omega^n = 1$. So
$\sum_{k=0}^{n-1} \omega^k = \frac{\omega^n - 1}{\omega - 1} = \frac{1-1}{\omega-1} = 0$

(c) (i) Find all $\omega$ such that $\omega^2 = -1$.

**Solution:** $\omega = i, -i$

There are many ways to arrive at the solution, here's one: Squaring both sides, we get $\omega^4 = 1$. So we only need to consider the 4th roots of unity, $e^{2\pi i \cdot 0/4}, e^{2\pi i \cdot 1/4}, e^{2\pi i \cdot 2/4}, e^{2\pi i \cdot 3/4}$, or equivalently $1, i, -1, -i$. Geometrically, we get these by going $0, 1, 2, 3$ quarters of the way around the complex unit circle. Of these four values, the ones that square to $-1$ are $i, -i$.

(ii) Find all $\omega$ such that $\omega^4 = -1$.

**Solution:** $\omega = e^{2\pi i \cdot 1/8}, e^{2\pi i \cdot 3/8}, e^{2\pi i \cdot 5/8}, e^{2\pi i \cdot 7/8}$

Similarly to the previous part, squaring both sides we get $\omega^8 = 1$, so we only need to consider the 8th roots of unity. However, $\omega^4 \neq 1$, so $\omega$ is not a 4th root of unity. The 8th roots of unity that are not 4th roots of unity are $e^{2\pi i \cdot 1/8}, e^{2\pi i \cdot 3/8}, e^{2\pi i \cdot 5/8}, e^{2\pi i \cdot 7/8}$, and we can check that these all are solutions to $\omega^4 = -1$.