Note: Your TA may not get to all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. The discussion worksheet is also a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

# 1 Master Theorem

For solving recurrence relations asymptotically, it often helps to use the Master Theorem:

<table>
<thead>
<tr>
<th>Master Theorem. If $T(n) = aT(n/b) + O(n^d)$ for $a &gt; 0$, $b &gt; 1$, and $d \geq 0$, then</th>
</tr>
</thead>
</table>
| $T(n) = \begin{cases} 
O(n^d) & \text{if } d > \log_b a \\
O(n^{d \log n}) & \text{if } d = \log_b a \\
O(n^{\log_b a}) & \text{if } d < \log_b a 
\end{cases}$ |

Note: You can replace $O$ with $\Theta$ and you get an alternate (but still true) version of the Master Theorem that produces $\Theta$ bounds.

d_{\text{crit}} = \log_b a$ is called the critical exponent. Notice that whichever of $d_{\text{crit}}$ and $d$ is greater determines the growth of $T(n)$, except in the case where they are perfectly balanced.

Solve the following recurrence relations and give a $O$ bound for each of them.

(a) (i) $T(n) = 3T(n/4) + 4n$
(ii) $T(n) = 45T(n/3) + .1n^3$

(b) $T(n) = 2T(\sqrt{n}) + 3$, and $T(2) = 3$.
   
   Hint: Try repeatedly expanding the recurrence.

(c) Consider the recurrence relation $T(n) = 2T(n/2) + n \log n$. We can’t plug it directly into the Master Theorem, so solve it by adding the size of each layer.
   
   Hint: split up the $\log(n/(2^i))$ terms into $\log n - \log(2^i)$, and use the formula for arithmetic series.
2 Sorted Array

Given a sorted array $A$ of $n$ (possibly negative) distinct integers, you want to find out whether there is an index $i$ for which $A[i] = i$. Devise a divide-and-conquer algorithm that runs in $O(\log n)$ time.
3 Quantiles

Let $A$ be an array of length $n$. The boundaries for the $k$ quantiles of $A$ are \( \{ a^{(n/k)}, a^{(2n/k)}, \ldots, a^{((k-1)n/k)} \} \) where $a^{(\ell)}$ is the $\ell$-th smallest element in $A$.

Devise an algorithm to compute the boundaries of the $k$ quantiles in time $O(n \log k)$. For convenience, you may assume that $k$ is a power of 2.

*Hint:* Recall that $\text{QUICKSELECT}(A, \ell)$ gives $a^{(\ell)}$ in $O(n)$ time.
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:

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

In polar form, \( r \geq 0 \) represents the distance of the complex number from 0, and \( \theta \) represents its angle. The \( n \) roots of unity are the \( n \) complex numbers satisfying \( \omega^n = 1 \). They are given by

\[
\omega_k = e^{2\pi ik/n}, \quad k = 0, 1, 2, \ldots, n-1
\]

(a) Let \( \omega_1 = e^{2\pi i 3/10}, \omega_2 = e^{2\pi i 5/10} \) be two 10-th roots of unity. Compute the product \( \omega_1 \cdot \omega_2 \). Is this a root of unity? Is it an 10-th root of unity?

What happens if \( \omega_1 = e^{2\pi i 6/10}, \omega_2 = e^{2\pi i 7/10} \)?

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

*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!

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

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