Here are some extra practice problems taken from past semesters. You are welcomed to check your solutions with me or discuss them in the telegram group chat.

1. **Asymptotic Analysis** (AY 22/23 Sem 2). State with proof whether the following statements are TRUE or FALSE.
   (a) Suppose $f(n) = n(1 - \sin n)$ (defined over non-negative reals). Then $f(n) \notin O(n)$.
   (b) Suppose $f(n) = n(1 - \sin n)$ (defined over non-negative reals). Then $f(n) \notin \Omega(n)$.
   (c) Let $T(n) = 7T(n/7) + f(n)$ (with the base case being $T(1) = \Theta(1)$). If $f(n) \in O(n \log n)$, then $T(n) \in \Omega(n \log n)$.

2. **Recurrences** (AY 24/25 Sem 2). Solve the following recurrence relations. Provide as tight a bound as possible. If you use the master theorem, state the case that applies, along with a short reasoning why the case applies. Otherwise, give a detailed proof (using any of the methods). Unless otherwise specified, you can assume base cases, $T(n)$ for $n \leq$ some constant, to be $\Theta(1)$.
   (a) $T(n) = 6T(n/3) + n^2$.
   (b) $T(n) = 9T(n/2) + 6n^3 + 4$.
   (c) $T(n) = T(n/7) + T(6n/7) + 5$. (Assume $7 \leq T(n) \leq 100$ for $n \leq 7$.)

3. **Recurrences** (AY 23/24 Sem 2 Practice Set). For each of these, try figuring out why you can't use master theorem to solve these recurrences, you do not need to be formal. Next, get (with proof) as tight bound as possible, for both upper and lower.
   (a) $T(n) = 2T(n-1) + \Theta(1)$
   (b) $T(n) = T(\sqrt{n}) + \Theta(1)$
   (c) $T(n) = 2T(\sqrt{n}) + \Theta(1)$
   (d) $T(n, m) = T(\frac{n}{2}, m) + \Theta(m)$
   (e) $T(n) = (\sqrt{n} + 1)T(\sqrt{n}) + \sqrt{n}$

4. **Recurrences** (AY 19/20 Sem 2). Solve the following recurrence relations. Provide as tight a bound as possible. If you use the master theorem, state the case that applies, along with a short reasoning why the case applies. Otherwise, give a detailed proof (using any of the methods). Unless otherwise specified, you can assume base cases, $T(n)$ for $n \leq$ some constant, to be $\Theta(1)$.
   (a) $T(n) = 9T(n/3) + n^2/\log n$.
   (b) $T(2^n) = T(2^0) + T(2^1) + \cdots + T(2^{n-1}) + n^2$.
   (c) $T(n) = 5T(n/4) + n^4/\log\log n$.
   (d) $T(n) = 4T(n/4) + n \log\log\log n$.

5. **Proof of Correctness** (AY 23/24 Sem 2 Practice Set). Consider the function Fun(x, y). What is its output? What is the invariant for the while loop? Can you show that this algorithm correctly compute the output?

Algorithm 1

```
1: procedure FUN(x, y)
2:     ans ← 0, p ← x, q ← y
3:     while q > 0 do
4:         r ← q mod 2
5:         q ← q/2
6:         ans ← ans + r × p
7:         p ← p × 2
8:     return ans
```

6. **Divide and Conquer, Proof of Correctness** (AY 24/25 Sem 2, Modified).
   You are consulting for the NUS Office of Student Affairs (OSA), which is investigating cheating cases involving bootleg, defaced student cards. The OSA has confiscated a collection of n student cards suspected of being unauthorized duplicates. Each student card is a small plastic card embedded with a chip that securely stores encrypted data. Each card's encrypted data correspond to a unique student (student number), and multiple student cards may correspond to the same student number. Since the cards are defaced, it is difficult to read the student number directly from the cards. Instead, OSA uses a 'matching device', a device capable of determining whether two student cards correspond to the same student number after performing some computations.
   The OSA poses the following question: among the collection of n confiscated cards, is there a subset of more than $n/2$ cards that all correspond to the same student number?
   Due to security concerns, you are not able to know the student number on a card, but are only allowed to interact with the matching device. Specifically, the only operation you can perform is to select two student cards and use the matching device to determine whether they correspond to the same student number.

   (a) Design an efficient method to aid OSA's investigation using $O(n \log n)$ queries to the matching device. Analyze the query complexity of your algorithm (i.e., the number of queries made by your algorithm in terms of n).

   (b) Give a formal proof of correctness on why your algorithm is correct.

   (Note: There is a recursive algorithm that takes $\Theta(n \log n)$ queries, and an iterative algorithm that takes $\Theta(n)$ queries. You are welcomed to try both.)

7. **Divide and Conquer** (AY 23/24 Sem 2 Practice Set). Given an array A of n integers (possibly 0 or negative as well), find the largest possible value c that can be obtained by summing up the values in some contiguous subarray of A, i.e., $c = A[i] + A[i+1] + A[i+2] + \cdots + A[i+t]$ for some i, t. Think of a divide and conquer solution that does it in $O(n \log n)$ time. As a bonus, you can then think about how to improve it to $O(n)$ by some minor modifications.

8. **Lower bounds** (AY 23/24 Sem 2 Practice Set, Modified). Consider an array of distinct integers sorted in increasing order. The array has then been rotated (anti-clockwise) k number of times, i.e., all the numbers in the sorted array have been (cyclically) shifted k places on the leftside. Now given such an array, find the value of k. Prove that your algorithm is optimal (asymptotically).

9. **Lower bounds** (New Question). There are N students $\{1, 2, \ldots, N\}$ interning at N companies $\{1, 2, \ldots, N\}$. You know that each student interns at exactly one company, and no two students intern at the same company. Each student knows which company he/she is ownself interning at, but there is no information about the other students. For each round, you may ask any student a Yes/No question about the company the student interns at.

   (a) Design an algorithm that, with as few Yes/No questions as possible, finds out the mapping between the students and the companies. Your algorithm only has to be asymptotically optimal (multiplicative and constant factors do not matter).

   (b) Prove that your algorithm is optimal asymptotically, i.e. any correct algorithm must take, asymptotically, at least as many Yes/No questions.

Answers

1. (a) FALSE (b) TRUE (c) FALSE
2. (a) $\Theta(n^2)$ (b) $\Theta(n^{\log_2 9})$ (c) $\Theta(n)$
3. (a) $\Theta(2^n)$ (b) $\Theta(\log \log n)$ (c) $\Theta(\log n)$ (d) $\Theta(m \log n)$ (e) $\Theta(n)$
4. (a) $\Theta(n^2 \log \log n)$ (b) $T(2^n) = \Theta(2^n)$ (c) $\Theta(n^4 / \log \log n)$ (d) $\Theta(n \log n \log \log \log n)$
5. $x \times y$
8. $O(\log n)$
9. $O(n \log n)$