1) Give a Θ estimate for each of the following recurrences (1), (2) and (3), for $n > 1$:

$$T_1(n) = T_1(\tfrac{n}{3}) + c \quad (1)$$
$$T_2(n) = 7 \cdot T_2(\tfrac{n}{6}) + 3n^2 \quad (2)$$
$$T_3(n) = 32 \cdot T_3(\tfrac{n}{2}) + n^4 \quad (3)$$

Solution: The Master theorem applies to each of the three recurrences:

(1) since for $a = 1$, $b = 3$, $s = 0$ we have $a = 1 = 3^0 = b^s$, its 2nd case applies, so $T_1 \in \Theta(n^s \log n) = \Theta(\log n)$;

(2) since for $a = 7$, $b = 6$, $s = 2$ we have $a = 7 < 6^2 = b^s$, its 3rd case applies, so $T_2 \in \Theta(n^s) = \Theta(n^2)$;

(3) since for $a = 32$, $b = 2$, $s = 4$ we have $a = 32 > 2^4 = b^s$, its 1st case applies, so $T_3 \in \Theta(n^{\log_b a}) = \Theta(n^{\log_2 32}) = \Theta(n^5)$.

2) Consider binary trees which store natural numbers in their leaves, and a function which sums the leaf values, for example, in a Haskell-like notation:

```
sum (Leaf a) = a
sum (Node l r) = (sum l) + (sum r)
```

Assume that the time complexity of function `+` is $\Theta(1)$.

a) Explain why the Master theorem can not be used to compute the time complexity of `sum` for an *arbitrary* tree.

b) Describe a class of trees which allows you to use the Master theorem to compute the time complexity of `sum`. Compute the complexity `sum` on this class of trees.

Solution:

a) The Master theorem requires the data to be split to several parts of the same size. Since the left sub-tree might have a different size than the right sub-tree, we can not directly apply it for a general binary tree.

b) We can, however, use the Master theorem provided the tree is balanced, that is, the left sub-tree of every node has the same size as its right sub-tree. Then, the time complexity can be described by the following recurrent equation:

$$T(n) = 2 \cdot T(\tfrac{n}{2}) + c$$

Hence the Master theorem states that $T \in \Theta(n)$.

3) Consider the following recurrence defined for $n = 2^k$ whenever $k > 0$.

$$T(n) = \begin{cases} \log_2(n) & \text{if } n \leq 4 \\ T(\frac{n}{2}) + 2 \cdot T(\frac{n}{4}) & \text{otherwise (if } n = 2^k) \end{cases}$$

Compute the values $T(n)$ for the first (smallest) 10 admissible values of n. Based on this observation, guess the closed form of T (a non-recurrent equivalent of T) and prove your guess to be correct.

Solution: The recurrence $T(n)$ is defined for $n = 2^k$ with $k > 0$, hence the first 10 admissible values of n are $2^1, 2^2, \dots, 2^{10}$. The first 10 values $T(n)$ are as follows.

n	2	4	8	16	32	64	128	256	512	1024
$T(n)$	1	2	4	8	16	32	64	128	256	512

From this, we can guess that $T(n) = n/2$. We can easily prove this by induction on n ordered by $\leq$. For $n = 2$ we have $T(2) = \log_2(2) = 1 = 2/2$. Similarly, for $n = 4$ we have $T(4) = \log_2(4) = 2 = 4/2$. For $n = 2^k > 4$ we have $T(n) = T(n/2) + 2 \cdot T(n/4) \stackrel{\text{IH}}{=} n/4 + 2 \cdot (n/8) = n/2$. Here, we also need to verify that the induction hypothesis is used correctly. The induction hypothesis holds for all $n_0 < n$ which are of the shape 2^{k_0} for some $k_0 > 0$. Since $n = 2^k$, we are using the hypothesis for $n_0 = n/2 = 2^{k-1}$ and for $n_0 = n/4 = 2^{k-2}$. That is, we are using the induction hypothesis for $k_0 = k - 1$ and for $k_0 = k - 2$. Since $n > 4$ we have $k > 2$ and thus $k_0 > 0$ holds in both the cases.

4) Consider the following recurrence $T : \mathbb{N} \to \mathbb{N}$:

$$T(0) = 0 \qquad\qquad T(n) = T(\lfloor \tfrac{n}{3} \rfloor) + T(\lfloor \tfrac{n}{4} \rfloor) + n \quad \text{for } n \geq 1$$

What can you say about the asymptotic growth of T?

Solution: The Master Theorem does not apply here since we partition the input into two different sizes. However, we can still derive the asymptotic growth of $T(n)$ with some tricks.

It is obvious from the definition that T is non-negative and strictly increasing, and in fact $T(n) \geq n$ (proof by induction). We thus have $T(n) \in \Omega(n)$.

Next, we try to bound $T(n)$ from above. Since $T(n)$ is increasing, we can bound it by the following simpler recurrence T':

$$T'(0) = 0 \qquad\qquad T'(n) = 2T'(\lfloor \tfrac{n}{3} \rfloor) + n \quad \text{for } n \geq 1$$

A straightforward induction shows that $T(n) \leq T'(n)$ for all n. Moreover, T' falls within the scope of the Master Theorem with $s = 1$ (third case), which tells us that $T'(n) \in \Theta(n^s) = \Theta(n)$, and therefore $T(n) \in O(n)$. We can then conclude that $T(n) \in \Theta(n)$.

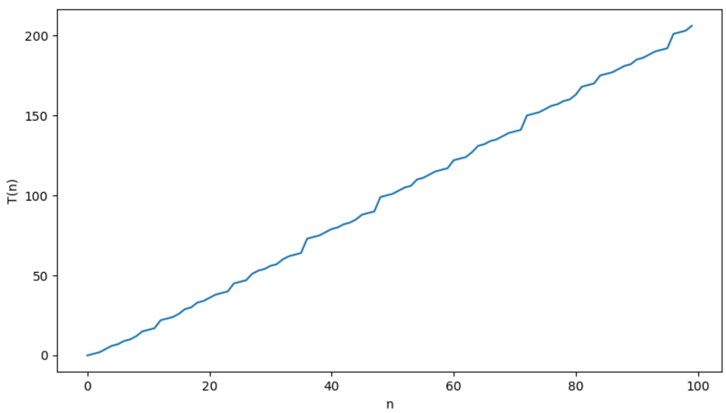

Figure 1: A plot of the recurrence $T(n)$ from exercise 4 that illustrates its linear growth.

A more direct approach: One can also directly prove $T(n) \leq \frac{12}{5}n$ by a straightforward induction: the base case $n = 0$ is obvious. For the induction step, we note that

$$T(n) \;=\; T(\lfloor \tfrac{n}{3} \rfloor) + T(\lfloor \tfrac{n}{4} \rfloor) + n \;\overset{\text{IH}}{\leq}\; \tfrac{12}{5}\lfloor \tfrac{n}{3} \rfloor + \tfrac{12}{5}\lfloor \tfrac{n}{4} \rfloor + n \;\leq\; \tfrac{12}{5} \cdot \tfrac{n}{3} + \tfrac{12}{5} \cdot \tfrac{n}{4} + n \;=\; \tfrac{12}{5}n$$

and we are done.

But how does one come up with the magical constant $\frac{12}{5}$ here? For this, we can simply start with the *ansatz* $T(n) \leq cn$ and then attempt to do the above induction proof until we get stuck. The place where we get stuck is the final one, where we must show that $(\frac{7}{12}c+1)n \leq cn$. Solving this for c, we then obtain $c \geq \frac{12}{5}$, which suggests that choosing $c = \frac{12}{5}$ makes the proof go through – and indeed it does. As always, coming up with the right *ansatz* is a matter of experience, but looking at a plot (see Figure **??**) can help.

For the unusually curious student: If we had had $\frac{n}{2}$ instead of $\frac{n}{4}$ in the recurrence, the Master Theorem approach would not have worked anymore since that would only have given us the estimate $T(n) \in O(n \log n)$, which is correct, but weaker than necessary. The direct approach would still have worked with the constant $c = 6$.

There is also a generalisation of the Master Theorem known as the *Akra–Bazzi Theorem* that can handle such recurrences directly.