CS5800: Algorithms — Spring '21 — Virgil Pavlu

Homework 1
Due : Wednesday, January 27 at 11:59pm via Gradescope

Name: Zerun Tian

Instructions:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- Please review the grading policy outlined in the course information page.
- You must also write down with whom you worked on the assignment. If this changes from problem to problem, then you should write down this information separately with each problem.
- Problem numbers (like Exercise 3.1-1) are corresponding to CLRS 3^{rd} edition. While the 2^{nd} edition has similar problems with similar numbers, the actual exercises and their solutions are different, so make sure you are using the 3^{rd} edition.

1. ***(20 points)***

Two linked lists (simple link, not double link) heads are given:headA, andheadB;it is also given that the two lists intersect, thus after the intersection they have thesame elements to the end. Find the first common element, without modifying the listselements or using additional datastructures.

(a) A linear algorithm is discussed in the lecture: count the lists first, then use the count difference as an offset in the longer list, before traversing the lists together. Write a formal pseudocode (the pseudocode in the lecture is vague), using "next" as a method/pointer to advance to the next element in a list.

Solution:

Counting the number of nodes in a linked list (helper function):

```
function COUNTNODES(head)
    count = 0
    p = head
    while p is not None do
        count = count + 1
        p = p.next
    return count
```

Finding the first intersecting node:

```
function FINDINTERSECTION(headA, headB)
    lenA = COUNTNODES(headA)
    lenB = COUNTNODES(headB)
    offset = abs(lenA − lenB)          ▷ difference in size between the two lists
    pl = headA                         ▷ a pointer to the longer list
    ps = headB                         ▷ a pointer to the shorter list
    if lenA < lenB then
        pl = headB
        ps = headA
    while offset > 0 do                ▷ move the pointer of the longer list by offset
        pl = pl.next
        offset = offset − 1
    while pl is not None do            ▷ iterate simultaneously to find the intersecting node
        if pl = ps then
            return pl
        pl = pl.next
        ps = ps.next
    return None                        ▷ no intersecting node found
```

(b) Write the actual code in a programming language (C/C++, Java, Python etc) of your choice and run it on a made-up test pair of two lists. A good idea is to use pointers to represent the list linkage.

Solution: Code is listed in the file `hw1.py`

2. ***(10 points)*** *Exercise 3.1-1*

Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

Solution:

By the definition of big-Θ, we are trying to show that with positive constants c_1, c_2 and n_0,

$$c_1 \cdot (f(n) + g(n)) \leq \max(f(n), g(n)) \leq c_2 \cdot (f(n) + g(n))$$

for $n \geq n_0$.

Since $f(n)$ and $g(n)$ are asymptotically nonnegative, there exists a positive constant n_f s.t. $f(n) \geq 0$ when $n \geq n_f$. Similarly, $g(n) \geq 0$ when $n \geq n_g$. We know that $f(n) + g(n)$ is nonnegative when $n \geq \max(n_f, n_g)$, which met one of the necessary conditions of big-Θ.

We can derive some other findings for $n \geq \max(n_f, n_g)$,

$$f(n) \leq \max(f(n), g(n))$$

$$g(n) \leq \max(f(n), g(n))$$

By combining the two inequalities above, we get $\frac{1}{2}(f(n) + g(n)) \leq \max(f(n), g(n))$. Moreover, we know $\max(f(n), g(n)) \leq f(n) + g(n)$ because they are nonnegative functions.

In conclusion, we find the formula below holds,

$$\frac{1}{2} \cdot (f(n) + g(n)) \leq \max(f(n), g(n)) \leq f(n) + g(n)$$

with $c_1 = \frac{1}{2}$, $c_2 = 1$, and $n_0 = max(n_f, n_g)$, so $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

3. ***(5 points)*** *Exercise 3.1-4*

Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

Solution:

The LHS 2^{n+1} can be written as $2 \cdot 2^n$. By definition of big-O, we want to find a constant c such that $2 \cdot 2^n \leq c \cdot 2^n$. For example, $c = 3$ is a valid choice. We showed $2^{n+1} = O(2^n)$.

The LHS 2^{2n} can be written as $(2^2)^n$, which is 4^n. We attempt to find a constant c such that $4^n \leq c \cdot 2^n$. Let's move the terms to get $\frac{4^n}{2^n} \leq c$. The simplified formula $2^n \leq c$ does not hold as exponential grows much faster than constant. We showed $2^{2n} \neq O(2^n)$.

4. ***(15 points)***

Rank the following functions in terms of asymptotic growth. In other words, find an arrangement of the functions $f_1, f_2, ...$ such that for all i, $f_i = \Omega(f_{i+1})$.

$$\sqrt{n}\ln n \quad \ln\ln n^2 \quad 2^{\ln^2 n} \quad n! \quad n^{0.001} \quad 2^{2\ln n} \quad (\ln n)!$$

Solution:

In decreasing order of growth rate:

$$n! \geq 2^{\ln^2 n} \geq (\ln n)! \geq 2^{2\ln n} \geq \sqrt{n}\ln n \geq n^{0.001} \geq \ln\ln n^2$$

Reasoning:

1. Show $n! = \Omega(2^{\ln^2 n})$:
   Using the Stirling's approximation, $n! \approx \sqrt{2\pi}n^{n+1/2}e^{-n}$. Let's simplify the RHS formula $2^{\ln^2 n}$,

   $$2^{\ln^2 n} = (2^{\ln n})^{\ln n} = (n^{\ln 2})^{\ln n}$$

   where $2^{\ln n}$ can be expressed as $n^{\ln 2}$ based on the formula $a^{\log_c^b} = b^{\log_c^a}$ we learned during the first tutoring session. Then, we can show that,

   $$\begin{aligned}\lim_{n\to\infty}\frac{n!}{2^{\ln^2 n}} \approx \lim_{n\to\infty}\frac{\sqrt{2\pi}n^{n+1/2}e^{-n}}{(n^{\ln 2})^{\ln n}} &= \lim_{n\to\infty}\frac{\ln n^{n+1/2} - n}{\ln 2\ln n\ln n}\\ &= \lim_{n\to\infty}\frac{(n+1/2)\ln n - n}{\ln 2\ln^2 n}\\ &= \lim_{n\to\infty}\frac{n\ln n + (\ln n)/2 - n}{\ln 2\ln^2 n}\\ &\approx \lim_{n\to\infty}\frac{n}{\ln 2\ln n}\\ &= \infty\end{aligned}$$

2. Show $2^{\ln^2 n} = \Omega((\ln n)!)$:
   We have that $(\ln n)! \approx \sqrt{2\pi}(\ln n)^{\ln n+1/2}e^{-\ln n}$ using the Stirling's approximation. We take the limit as follows,

   $$\begin{aligned}\lim_{n\to\infty}\frac{2^{\ln^2 n}}{(\ln n)!} \approx \lim_{n\to\infty}\frac{(n^{\ln 2})^{\ln n}}{\sqrt{2\pi}(\ln n)^{\ln n+1/2}e^{-\ln n}} &\approx \lim_{n\to\infty}\frac{(n^{\ln 2})^{\ln n}}{(\ln n)^{\ln n+1/2}e^{-\ln n}}\\ &= \lim_{n\to\infty}\frac{\ln 2\ln^2 n}{(\ln n+1/2)\ln\ln n - \ln n}\\ &= \lim_{n\to\infty}\frac{\ln 2\ln^2 n}{(\ln\ln n)\ln n + (\ln\ln n)/2 - \ln n}\\ &\approx \lim_{n\to\infty}\frac{\ln 2\ln n}{\ln\ln n - 1}\\ &= \infty\end{aligned}$$

3. Show $(\ln n)! = \Omega(2^{2\ln n})$:
   We take the limit as follows,

$$\begin{aligned}
\lim_{n\to\infty} \frac{(\ln n)!}{2^{2\ln n}} \approx \lim_{n\to\infty} \frac{(\ln n)^{\ln n + 1/2} \cdot e^{-\ln n}}{4^{\ln n}} &= \lim_{n\to\infty} \frac{(\ln n)^{\ln n} \cdot (\ln n)^{1/2} e^{-\ln n}}{4^{\ln n}} \\
&= \lim_{n\to\infty} \frac{(\ln\ln n)\ln n + (\ln\ln n)/2 - \ln n}{(\ln 4)\ln n} \\
&\approx \lim_{n\to\infty} \frac{(\ln\ln n) - 1}{\ln 4} \\
&= \infty
\end{aligned}$$

4. Show $2^{2\ln n} = \Omega(\sqrt{n}\ln n)$:
   The LHS can be written as $2^{2\ln n} = 4^{\ln n} = n^{\ln 4} \approx n^{1.386}$. We take the limit as follows,

$$\lim_{n\to\infty} \frac{2^{2\ln n}}{\sqrt{n}\ln n} \approx \lim_{n\to\infty} \frac{n^{\ln 4}}{n^{0.5}\ln n} = \lim_{n\to\infty} \frac{n^{((\ln 4)-0.5)}}{\ln n} = \infty$$

5. Show $\sqrt{n}\ln n = \Omega(n^{0.001})$:
   Let's take the limit as follows,

$$\lim_{n\to\infty} \frac{\sqrt{n}\ln n}{n^{0.001}} = \lim_{n\to\infty} \frac{n^{0.5}\ln n}{n^{0.001}} = \lim_{n\to\infty} \frac{n^{0.499}\ln n}{1} = \infty$$

6. Show $n^{0.001} = \Omega(\ln\ln n^2)$:
   We take the limit as follows,

$$\lim_{n\to\infty} \frac{n^{0.001}}{\ln\ln n^2} = \lim_{n\to\infty} \frac{n^{0.001}}{\ln(2\ln n)} = \lim_{n\to\infty} \frac{n^{0.001}}{\ln 2 + \ln\ln n} = \infty$$

5. ***(40 points)*** *Problem 4-1 (page 107)*

Give asymptotic upper and lower bounds for T(n) in each of the following recurrences. Assume that T(n) is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your answers.

(a) $T(n) = 2T(n/2) + n^4$

Solution:

It has the form $T(n) = aT(n/b) + n^c$, so we use the master method learned in class.

The three cases are:

- case 1: $\log_b^a > c$, then $T(n) = \Theta(n^{\log_b^a})$
- case 2: $\log_b^a = c$, then $T(n) = \Theta(n^{\log_b^a} \log n)$
- case 3: $\log_b^a < c$, then $T(n) = \Theta(n^c)$

With $a = 2$, $b = 2$, and $c = 4$, we find $\log_b^a = \log_2^2 = 1 < c$.

This matches case 3, so we get that $T(n) = \Theta(n^4)$.

(b) $T(n) = T(7n/10) + n$

Solution:

With $a = 1$, $b = \frac{10}{7}$, and $c = 1$, we find $\log_b^a = \log_{10/7}^1 = 0 < c$.

This matches case 3, so we get that $T(n) = \Theta(n)$.

(c) $T(n) = 16T(n/4) + n^2$

Solution:

With $a = 16$, $b = 4$, and $c = 2$, we find $\log_b^a = \log_4^{16} = 2 = c$.

This matches case 2, so we get that $T(n) = \Theta(n^2 \log n)$.

(d) $T(n) = 7T(n/3) + n^2$

Solution:

With $a = 7$, $b = 3$, and $c = 2$, we find $\log_b^a = \log_3^7 \approx 1.771 < c$.

This matches case 3, so we get that $T(n) = \Theta(n^2)$.

(e) $T(n) = 7T(n/2) + n^2$

Solution:

With $a = 7$, $b = 2$, and $c = 2$, we find $\log_b^a = \log_2^7 \approx 2.807 > c$.

This matches case 1, so we get that $T(n) = \Theta(n^{\log_2^7})$.

(f) $T(n) = 2T(n/4) + \sqrt{n}$

Solution:

With $a = 2$, $b = 4$, and $c = 0.5$, we find $\log_b^a = \log_4^2 = 0.5 = c$.

This matches case 2, so we get that $T(n) = \Theta(\sqrt{n} \log n)$.

(g) $T(n) = T(n-2) + n^2$

Solution:

Let's solve this recurrence via iteration.

$$\begin{aligned}
T(n) &= T(n-2) + n^2 \\
&= [T(n-4) + (n-2)^2] + n^2 = T(n-4) + (n-2)^2 + n^2 \\
&= [T(n-6) + (n-4)^2] + (n-2)^2 + n^2 = T(n-6) + (n-4)^2 + (n-2)^2 + n^2 \\
&\vdots \\
&= T(n-2k) + (n-2(k-1))^2 + \cdots + (n-4)^2 + (n-2)^2 + (n-0)^2 \\
&= T(n-2k) + \sum_{i=0}^{k-1} (n-2i)^2
\end{aligned}$$

Now, we show this pattern of k is correct, by induction.

Claim: For all $k \geq 1$, $T(n-2k) + \sum_{i=0}^{k-1} (n-2i)^2$.

Proof:

- The base case, $k = 1$, is true as the resulting equation $T(n-2) + n^2$ matches the original recurrence.

- Inductive hypothesis: assuming the claim is true for $k = j$. i.e.,

$$T(n) = T(n-2j) + \sum_{i=0}^{j-1} (n-2i)^2$$

- Inductive step: showing the claim holds true for $k = j+1$. i.e.,

$$T(n) = T(n-2(j+1)) + \sum_{i=0}^{j} (n-2i)^2$$

Let's expand the inductive hypothesis by applying the definition of the recurrence,

$$\begin{aligned}
T(n) &= T(n-2j) + \sum_{i=0}^{j-1} (n-2i)^2 \\
&= [T(n-2j-2) + (n-2j)^2] + \sum_{i=0}^{j-1} (n-2i)^2 \\
&= T(n-2(j+1)) + (n-2j)^2 + (n-2(j-1))^2 + \cdots + (n-2)^2 + (n-0)^2 \\
&= T(n-2(j+1)) + \sum_{i=0}^{j} (n-2i)^2
\end{aligned}$$

We thus have that $T(n) = T(n-2k) + \sum_{i=0}^{k-1} (n-2i)^2$ for $k \geq 1$. Now, let's choose a k that would lead to a base case. Given $T(n)$ is constant for $n \leq 2$, we want $n - 2k = 2$, then $k = \frac{n}{2} - 1$,

$$
\begin{aligned}
T(n) &= T(2) + \sum_{i=0}^{n/2-2} (n-2i)^2 \\
&= T(2) + \sum_{i=0}^{n/2-2} n^2 - 4ni + 4i^2 \\
&= T(2) + n^2 \sum_{i=0}^{n/2-2} 1 - 4n \sum_{i=0}^{n/2-2} i + 4 \sum_{i=0}^{n/2-2} i^2 \\
&= T(2) + n^2(\frac{n}{2} - 1) - 4n(\frac{1}{2}(1 + \frac{n}{2} - 2)(\frac{n}{2} - 2)) + 4(\frac{1}{6}(\frac{n}{2} - 2)(\frac{n}{2} - 2 + 1)(n - 4 + 1)) \\
&= T(2) + \frac{n^3}{2} - n^2 - \frac{n^3}{2} + 2n^2 + n^2 - 4n + \frac{n^3}{6} - n^2 + \frac{4n}{3} - \frac{n^2}{2} + 3n - 4 \\
&= T(2) + \frac{n^3}{6} - \frac{n^2}{2} - \frac{n}{3} - 4 \\
&= \Theta(1) + \Theta(n^3) \\
&= \Theta(n^3)
\end{aligned}
$$

6. ***(30 points)*** *Problem 4-3 from (a) to (f) (page 108)*

Give asymptotic upper and lower bounds for T(n) in each of the following recurrences. Assume that T(n) is constant for sufficiently small n. Make your bounds as tight as possible, and justify your answers.

(a) $T(n) = 4T(n/3) + n\lg n$

Solution:

Using the iteration method, we expand the above recurrence as follows,

$$\begin{aligned}
T(n) &= 4T(\frac{n}{3}) + n\lg n \\
&= 4[4T(\frac{n}{9}) + \frac{n}{3}\lg\frac{n}{3}] + n\lg n = 16T(\frac{n}{9}) + 4\frac{n}{3}\lg\frac{n}{3} + n\lg n \\
&= 16[4T(\frac{n}{27}) + \frac{n}{9}\lg\frac{n}{9}] + 4\frac{n}{3}\lg\frac{n}{3} + n\lg n = 64T(\frac{n}{27}) + 4^2\frac{n}{9}\lg\frac{n}{9} + 4\frac{n}{3}\lg\frac{n}{3} + n\lg n \\
&= 64[4T(\frac{n}{81}) + \frac{n}{27}\lg\frac{n}{27}] + 4^2\frac{n}{9}\lg\frac{n}{9} + 4\frac{n}{3}\lg\frac{n}{3} + n\lg n \\
&= 256T(\frac{n}{81}) + 4^3\frac{n}{3^3}\lg\frac{n}{3^3} + 4^2\frac{n}{3^2}\lg\frac{n}{3^2} + 4\frac{n}{3}\lg\frac{n}{3} + n\lg n
\end{aligned}$$

Now, we can see a clear pattern in terms of k,

$$T(n) = 4^k T(\frac{n}{3^k}) + \sum_{i=0}^{k-1} 4^i \frac{n}{3^i}\lg\frac{n}{3^i}$$

We omit proving the pattern's correctness as the form is evident through several iterations. To reach a base case, we want to make $k \simeq log_3^n$. Plugging k into the recurrence gives us,

$$\begin{aligned}
T(n) &= 4^{\log_3^n} T(\frac{n}{3^{\log_3^n}}) + \sum_{i=0}^{\log_3^n - 1} 4^i \frac{n}{3^i}\lg\frac{n}{3^i} \\
&= n^{\log_3^4} T(1) + n\sum_{i=0}^{\log_3^n - 1} \frac{4^i}{3^i}(\lg n - \lg 3^i) \\
&= n^{\log_3^4} T(1) + n\sum_{i=0}^{\log_3^n - 1} \frac{4^i}{3^i}\lg n - n\lg 3\sum_{i=0}^{\log_3^n - 1} i(\frac{4}{3})^i
\end{aligned}$$

Let's try to resolve the second term and the third term of the above equation. The second term can be simplified further to,

$$\begin{aligned}
n\sum_{i=0}^{\log_3^n - 1} \frac{4^i}{3^i}\lg n &= n[\lg n + \frac{4}{3}\lg n + \frac{16}{9}\lg n + \cdots + (\frac{4}{3})^{\log_3^n - 1}\lg n] \\
&= n[\lg n \sum_{i=0}^{\log_3^n - 1} (\frac{4}{3})^i] \\
&= n[\lg n(3n^{\log_3^4 - 1} - 3)] \\
&= 3n^{\log_3^4}\lg n - 3n\lg n
\end{aligned}$$

Before we proceed on simplifying the third term, we know the closed form of the summation $\sum_{i=0}^{n} ia^i = \frac{a-a^{n+1}}{(1-a)^2} - \frac{na^{n+1}}{1-a}$. Thus, let's work on the third term,

$$
\begin{aligned}
n\lg 3 \sum_{i=0}^{\log_3^n -1} i(\frac{4}{3})^i &= n\lg 3(\frac{4/3-(4/3)^{\log_3^n}}{(1-4/3)^2} - \frac{(\log_3^n -1)(4/3)^{\log_3^n}}{1-4/3}) \\
&= n\lg 3(12 - 9n^{\log_3^4 -1} + 3n^{\log_3^4 -1}\log_3^n - 3n^{log_3^4 -1}) \\
&= (12\lg 3)n - (9\lg 3)n^{\log_3^4} + (3\lg 3)n^{\log_3^4}\log_3^n - (3\lg 3)n^{\log_3^4}
\end{aligned}
$$

Let's simplify the term $(3\lg 3)n^{\log_3^4}\log_3^n$ of the above equation,

$$
\begin{aligned}
(3\lg 3)n^{\log_3^4}\log_3^n &= 3n^{\log_3^4}\lg 3\log_3^n \\
&= 3n^{\log_3^4}\log_3^{n^{\lg 3}} \\
&= 3n^{\log_3^4}\log_3^{3^{\lg n}} \\
&= 3n^{\log_3^4}\lg n
\end{aligned}
$$

Combining the closed forms of the two summations, we write the recurrence as,

$$
\begin{aligned}
T(n) &= n^{\log_3^4}T(1) + n\sum_{i=0}^{\log_3^n -1}\frac{4^i}{3^i}\lg n - n\lg 3\sum_{i=0}^{\log_3^n -1} i(\frac{4}{3})^i \\
&= n^{\log_3^4}T(1) + [3n^{\log_3^4}\lg n - 3n\lg n] - [(12\lg 3)n - (9\lg 3)n^{\log_3^4} + 3n^{\log_3^4}\lg n - (3\lg 3)n^{\log_3^4}] \\
&= n^{\log_3^4}T(1) - 3n\lg n - (12\lg 3)n + (9\lg 3)n^{\log_3^4} + (3\lg 3)n^{\log_3^4} \\
&= [\Theta(1) + 9\lg 3 + 3\lg 3]n^{\log_3^4} - 3n\lg n - (12\lg 3)n \\
&= \Theta(n^{\log_3^4})
\end{aligned}
$$

Therefore, the asymptotic runtime is $T(n) = \Theta(n^{\log_3^4})$.

(b) $T(n) = 3T(n/3) + n/\lg n$

Solution:

Using the iteration method, we expand the above recurrence as follows,

$$
\begin{aligned}
T(n) &= 3T(\frac{n}{3}) + n/\lg n \\
&= 3[3T(\frac{n}{9}) + \frac{n}{3}(\lg\frac{n}{3})^{-1}] + n(\lg n)^{-1} = 9T(\frac{n}{9}) + n(\lg\frac{n}{3})^{-1} + n(\lg n)^{-1} \\
&= 9[3T(\frac{n}{27}) + \frac{n}{9}(\lg\frac{n}{9})^{-1}] + n(\lg\frac{n}{3})^{-1} + n(\lg n)^{-1} = 27T(\frac{n}{27}) + n(\lg\frac{n}{9})^{-1} + n(\lg\frac{n}{3})^{-1} + n(\lg n)^{-1}
\end{aligned}
$$

Now, we can see a clear pattern to represent the recurrence in terms of k,

$$
T(n) = 3^k T(\frac{n}{3^k}) + n\sum_{i=0}^{k-1}(\lg\frac{n}{3^i})^{-1}
$$

We omit proving the correctness of the pattern via induction because its form is evident. Then, we choose $k = \log_3^n$, which leads to a base case.

$$
\begin{aligned}
T(n) &= 3^{\log_3^n} T(\frac{n}{3^{\log_3^n}}) + n \sum_{i=0}^{\log_3^n - 1} (\lg \frac{n}{3^i})^{-1} \\
&= nT(1) + n[(\lg n)^{-1} + (\lg \frac{n}{3})^{-1} + (\lg \frac{n}{9})^{-1} + \cdots + (\lg \frac{n}{3^{\log_3^{n/3}}})^{-1}] \\
&= nT(1) + n[(\lg n)^{-1} + (\lg \frac{n}{3})^{-1} + (\lg \frac{n}{9})^{-1} + \cdots + (\lg 3)^{-1}]
\end{aligned}
$$

Here, we notice that the sequence in the bracket is a harmonic series, $\frac{1}{\lg 3}, \frac{1}{\lg 9}, \cdots, \frac{1}{\lg n/9}, \frac{1}{\lg n/3}, \frac{1}{\lg n}$. We can sum them up using the formula learned during the first tutoring session.

$$
\begin{aligned}
T(n) &= nT(1) + n \sum_{i=1}^{\log_3^n} \frac{1}{\lg 3^i} \\
&= \Theta(n) + \Theta(n \lg \lg n) \\
&= \Theta(n \lg \lg n)
\end{aligned}
$$

Finally, the asymptotic runtime is $T(n) = \Theta(n \lg \lg n)$.

(c) $T(n) = 4T(n/2) + n^2 \sqrt{n}$

Solution:

Let's simplify the $f(n)$ part of the equation,

$$T(n) = 4T(n/2) + n^2 \cdot n^{0.5} = 4T(n/2) + n^{2.5}$$

This recurrence has the form $T(n) = aT(n/b) + n^c$, so we apply the master theorem.
$a = 4$, $b = 2$, and $c = 2.5$. Let's do a test, $\frac{a}{b^c} = \frac{4}{2^{2.5}} < 1$, and it corresponds to the case 3 we learned in class. Thus, the asymptotic runtime is $T(n) = \Theta(n^{2.5})$.

(d) $T(n) = 3T(n/3 - 2) + n/2$

Solution:

The -2 of $T(n/3 - 2)$ can be omitted when we consider the asymptotic runtime because the change $n/3$ is more significant than the constant -2 for a large n. Then, the problem can be solved via the master method.
$a = 3$, $b = 3$, and $c = 1$. Let's do the test, $\frac{a}{b^c} = \frac{3}{3^1} = 1$, which corresponds to the case 2 we learned in class. Thus, the asymptotic runtime is $T(n) = \Theta(n^{\log_b^a} \log n) = \Theta(n \log n)$.

(e) $T(n) = 2T(n/2) + n/\lg n$

Solution:

Using the iteration method, we expand the above recurrence as follows,

$$
\begin{aligned}
T(n) &= 2T(\frac{n}{2}) + n/\lg n \\
&= 2[2T(\frac{n}{4}) + \frac{n}{2}(\lg \frac{n}{2})^{-1}] + n(\lg n)^{-1} = 4T(\frac{n}{4}) + n(\lg \frac{n}{2})^{-1} + n(\lg n)^{-1} \\
&= 4[2T(\frac{n}{8}) + \frac{n}{4}(\lg \frac{n}{4})^{-1}] + n(\lg \frac{n}{2})^{-1} + n(\lg n)^{-1} = 8T(\frac{n}{8}) + n(\lg \frac{n}{4})^{-1} + n(\lg \frac{n}{2})^{-1} + n(\lg n)^{-1}
\end{aligned}
$$

Now, we can see a clear pattern to represent this recurrence in terms of k,

$$T(n) = 2^k T(\frac{n}{2^k}) + n \sum_{i=0}^{k-1} (\lg \frac{n}{2^i})^{-1}$$

We omit proving its correctness via induction because its form is evident through the expansions. Then, we choose $k = \log_2^n$, which leads to a base case.

$$\begin{aligned}
T(n) &= 2^{\log_2^n} T(\frac{n}{2^{\log_2^n}}) + n \sum_{i=0}^{\log_2^n - 1} (\lg \frac{n}{2^i})^{-1} \\
&= nT(1) + n[(\lg n)^{-1} + (\lg \frac{n}{2})^{-1} + (\lg \frac{n}{4})^{-1} + \cdots + (\lg \frac{n}{2^{\log_2^{n/2}}})^{-1}] \\
&= nT(1) + n[(\lg n)^{-1} + (\lg \frac{n}{2})^{-1} + (\lg \frac{n}{4})^{-1} + \cdots + (\lg 2)^{-1}]
\end{aligned}$$

Similar to (b), what's inside the bracket is a harmonic series. The sum of a harmonic series is $\sum_{i=1}^{n} 1/i \sim \log_e^n + k$.

$$\begin{aligned}
T(n) &= nT(1) + n \sum_{i=1}^{\log_2^n} \frac{1}{\lg 2^i} \\
&= \Theta(n) + \Theta(n \lg \lg n) \\
&= \Theta(n \lg \lg n)
\end{aligned}$$

Thus, the asymptotic runtime is $T(n) = \Theta(n \lg \lg n)$.

(f) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

Solution:

Let's use the substitution method for this problem.

We guess $T(n) = \Theta(n)$, so we need to show $c_1 n \leq T(n) \leq c_2 n$.

Proof:

The lower bound is trivial to prove. Since the original recurrence has a term $+n$, we can choose $c1$ to be 0.5, then $0.5n \leq T(n/2) + T(n/4) + T(n/8) + n$.

Now, let's prove the upper bound is correct by induction. There are three hypotheses:

$$\begin{cases} T(n/2) \leq cn/2 \\ T(n/4) \leq cn/4 \\ T(n/8) \leq cn/8 \end{cases}$$

$$\begin{aligned}
T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + T(\frac{n}{8}) + n &\leq c\frac{n}{2} + c\frac{n}{4} + c\frac{n}{8} + n \\
&= (\frac{c}{2} + \frac{c}{4} + \frac{c}{8} + 1)n
\end{aligned}$$

We want,

$$(\frac{c}{2} + \frac{c}{4} + \frac{c}{8} + 1)n \overset{?}{\leq} cn$$

Say we choose c to be 16. $(8 + 4 + 2 + 1)n = 15n \leq 16n$. Thus, the guessed asymptotic runtime $T(n) = \Theta(n)$ is correct.