Homework 2

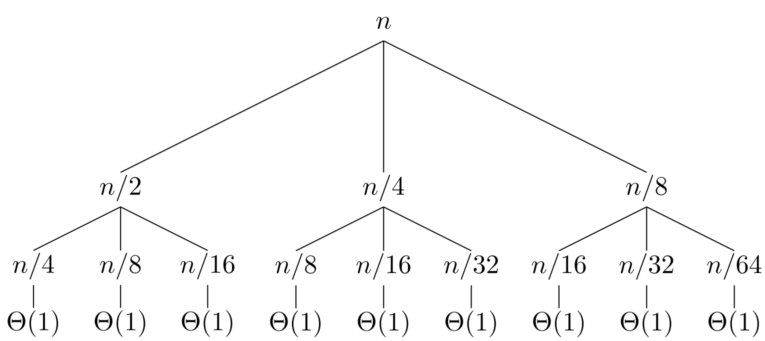
Walker Bagley

February 8, 2024

1. **Q1-2.3-8:** Loop over the set S of n integers while maintaining a set of seen numbers. For each element e, calculate the difference $x - e$ and check if that difference is contained in the seen set. If so, return true. Otherwise, add the current element e to the seen set and continue to the next element of set S.

2.
$$
\begin{aligned}
T(n) &= 5T(n/3) + n\lg n \\
\frac{a \cdot f(\frac{n}{b})}{f(n)} &= \frac{5 \cdot \frac{n}{3}\lg(\frac{n}{3})}{n \lg n} \\
&= \frac{5\lg(\frac{n}{3})}{3\lg n} = \frac{5(\lg n - \lg 3)}{3 \lg n} \\
&= \frac{5}{3}\left(1 - \frac{\lg 3}{\lg n}\right)
\end{aligned}
$$

The second term tends towards 0 as n gets very large, so the sum will approach 5/3 which is greater than 1. So, $T(n) = \Theta(n^{\log_3 5})$.

3.


$$
\begin{aligned}
0 &: \; n = \left(\frac{7}{8}\right)^0 n \\
1 &: \; \frac{7}{8}n = \left(\frac{7}{8}\right)^1 n \\
2 &: \; \frac{49}{64}n = \left(\frac{7}{8}\right)^2 n \\
k &: \; \left(\frac{7}{8}\right)^k n
\end{aligned}
$$

Worst case:

$$k = \log_2 n$$

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_2 n} \left(\frac{7}{8}\right)^i n = n \sum_{i=0}^{\log_2 n} \left(\frac{7}{8}\right)^i \\
&= n\left(\frac{1}{1 - \frac{7}{8}}\right) = 8n = O(n)
\end{aligned}
$$

Best case:

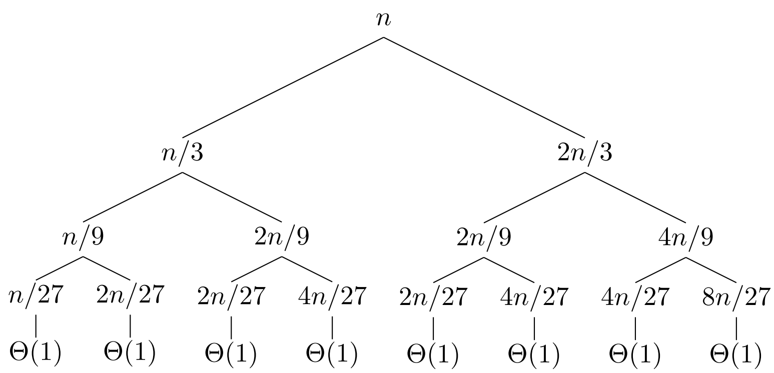
$$k = \log_8 n$$

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_8 n} \left(\frac{7}{8}\right)^i n = n \sum_{i=0}^{\log_8 n} \left(\frac{7}{8}\right)^i \\
&= n\left(\frac{1}{1 - \frac{7}{8}}\right) = 8n = \Omega(n)
\end{aligned}
$$

Worst Case Proof. Base case: $k = 0$. Then we have $T(n) = n = O(n)$. Assume for any depth $\leq k$ that we have $T(n) = O(n)$. Then for $k+1$, we have $T(n) = T(n/2) + T(n/4) + T(n/8) + n = n/2 + n/4 + n/8 + n = \frac{7}{8}n + n = \frac{15}{8}n = O(n)$. $\square$

Best Case Proof. Base case: $k = 0$. Then we have $T(n) = n = \Omega(n)$. Assume for any depth $\leq k$ that we have $T(n) = \Omega(n)$. Then for $k+1$, we have $T(n) = T(n/2) + T(n/4) + T(n/8) + n = n/2 + n/4 + n/8 + n = \frac{7}{8}n + n = \frac{15}{8}n = \Omega(n)$. $\square$

4.



In the worst case, we know that for depth k, $(\frac{2}{3})^k n = 1$, so $k = \log_{3/2} n = O(\log n)$. Each level in the tree also sums to n as all possible binomial combinations of $1/3$ and $2/3$ sum to 1 (this is a basic law of statistics as well). So there are $O(\log n)$ levels times n per level meaning that $T(n) = O(n \log n)$. The best case works similarly, with each level also summing to n. In this event, at depth k, $(\frac{1}{3})^k n = 1$, so $k = \log_3 n = O(\log n)$. Multiplying as before, $T(n) = \Omega(n \log n)$.