HW2 Solutions

1. #1 Solution
   Let M_w be the maximum number of nodes of weight $w.cost(T) = m(3)+n(1)$, where m and n will be non negative integers. To manually discover a series, we construct a tree and find: $M_0 = 1, M_1 = 1, M_2 = 1, M_3 = 2, M_4 = 3, M_5 = 4, M_6 = 6...$
   $\therefore we\ get\ M_w = M_{w-1} + M_{w-3}$

2. #2(i) Solution
   The problem forms the recursion $T(n) = 2T(n-1) + T(n-2) + 1$. The characteristic equation of the homogeneous part is

   $$r^2 = 2r + 1 \Rightarrow r^2 - 2r - 1 = 0 \Rightarrow r_1 = 1 + \sqrt{2}, r_2 = 1 - \sqrt{2}$$

   Further, the non-homogeneous part has the root $r = 1$, therefore the general solution is

   $$T(n) = A(1+\sqrt{2})^n + B(1-\sqrt{2})^n + C$$

   #2(ii) Solution
   Similarly, the characteristic root of homogeneous part is $r_1 = 4$, and the root of the non-homogeneous part is $r_2 = 2$. Therefore, the formula is

   $$T(n) = A4^n + B2^n$$

   Since $T(0) = 1$,$T(1) = 4T(0)+2^1 = 6$. Solving a simple linear equation yields $A = 2, B = -1$. Therefore, $T(n) = 24^n - 2^n$

   #2(iii) Solution
   The characteristic roots are $r_1 = r_2 = 4$, which are two repeated roots. Therefore, we have the formula

   $$T(n) = An4^n + B4^n$$

   Similarly, solving the linear equation yields $A = \frac{1}{2}, B = 2$.

3. #3 Solution

   $T(n) = [\sqrt{n} + c] + [\sqrt{n-c} + c] + [\sqrt{n-2c} + c] + ... + [\sqrt{2c} + c] + T(c)$

   It is easy to see the height of the recursion tree is $\frac{n}{c}$, and we have $\frac{n}{c}$ terms in the above sum. Then, it follows

   $$\frac{1}{2}\frac{n}{c}\sqrt{\frac{n}{2}} \leq \sqrt{\frac{n}{2}} + \sqrt{\frac{n}{2} + c} + \cdots + \sqrt{n} \leq \sum_{i=0}^{n/c} \sqrt{n - ic} \leq \frac{n}{c}\sqrt{n}$$

   Both $\frac{n}{2c}\sqrt{\frac{n}{2}}$ and $\frac{n}{c}\sqrt{n}$ grow at $\Theta(n\sqrt{n})$, therefore $T(n) = \Theta(n\sqrt{n} + \frac{n/c}{c}) = \Theta(n\sqrt{n})$.

4. #4 Solution

   $$\begin{aligned}
   T(n) &= f(n) + T(8n/9) + T(n/18) \\
   &= f(n) + f(8/9n) + f(1/18n) + T(8^2/9^2 n) + 2T(8/9 \cdot 1/18n) + T(1/18^2 n) \\
   &= f(n) + f(8/9n) + f(1/18n) + f(8^2/9^2 n) + 2f(8/9 \cdot 1/18n) + f(1/18^2 n) + \cdots \\
   &\cdots \\
   &= f(n) + f(8/9n) + f(1/18n) + f(8^2/9^2 n) + 2f(8/9 \cdot 1/18n) + f(1/18^2 n) \\
   &\quad + f(8^3/9^3 n) + 3f(8^2/9^2 \cdot 1/18n) + 3f(8/9 \cdot 1/18^2 n) + f(1/18^3 n) + \cdots
   \end{aligned}$$

   The recursion tree of this function is not perfectly balanced. The shortest height is $H_1 = \log_{18} n$, and the longest height is $H_2 \log_{9/8} n$.

   #4(i) When $f(n) = n$:

   In this case, we have

   $$\sum_{i=0}^{H_1} (8/9 + 1/18)^i n \leq T(n) \leq \sum_{i=0}^{H_2} (8/9 + 1/18)^i n$$

   Both sides are the sum of geometric sequence with common ratio $17/18 < 1$, therefore both of the sums are $\Theta(n)$. Therefore, $T(n) = \Theta(n)$.

   #4(ii) When $f(n) = n^2$:

   In this case, we have

   $$\sum_{i=0}^{H_1} (8^2/9^2 + 1^2/18^2)^i n^2 \leq T(n) \leq \sum_{i=0}^{H_2} (8^2/9^2 + 1^2/18^2)^i n^2$$

With the same reason as 4(i), both of the sums are $\Theta(n^2)$, and therefore $T(n) = \Theta(n^2)$.

5. #5 Solution

$$
\begin{aligned}
T(n) &= f(n) + T(8n/9) + T(n/6) \\
&= f(n) + f(8/9n) + f(1/6n) + T(8^2/9^2n) + 2T(8/9 \cdot 1/6n) + T(1/6^2n) \\
&= f(n) + f(8/9n) + f(1/6n) + f(8^2/9^2n) + 2f(8/9 \cdot 1/6n) + f(1/6^2n) + \cdots \\
&\cdots \\
&= f(n) + f(8/9n) + f(1/6n) + f(8^2/9^2n) + 2f(8/9 \cdot 1/6n) + f(1/6^2n) \\
&\quad + f(8^3/9^3n) + 3f(8^2/9^2 \cdot 1/6n) + 3f(8/9 \cdot 1/6^2n) + f(1/6^3n) + \cdots
\end{aligned}
$$

The recursion tree of this function is not perfectly balanced. The shortest height is $H_1 = \log_6 n$, and the longest height is $H_2 \log_{9/8} n$.

#5(i) When $f(n) = \log n$:

In this case, for the i-th level of the recursion tree (suppose the root is level-0), the work (or the time) we need is

$$
T_i(n) = \sum_{j=0}^{i} \left(\binom{i}{j} \log \left(\left(\frac{8}{9}\right)^{i-j} \left(\frac{1}{6}\right)^{j} n \right) \right) = (i-1)2^{i-2} \log \left(\frac{4}{27}\right) + 2^{i-1} \log n
$$

if i is less than the shortest height $H_1 = \log_6 n$.

Subsequently, the upper bound of the function can be derived by calculating the sum for $i \leq H_2$:

$$
\begin{aligned}
T(n) \leq \sum_{i=0}^{H_2} T_i(n) &= \sum_{i=0}^{H_2} \left(i2^{i-1} \log \frac{4}{27} + 2^i \log n \right) \\
&= \Theta(n^{\log_{9/8} 2} \log n)
\end{aligned}
$$

$$
\Rightarrow T(n) = O(n^{\log_{9/8} 2} \log n)
$$

Thus $T(n) = O(n^{\log_{9/8} 2} \log n)$.

#5(ii) When $f(n) = n^2$:

In this case, the problem is exactly same as 4(ii) except that the coefficients are different.

$$
\sum_{i=0}^{H_1} (8^2/9^2 + 1^2/6^2)^i n^2 \leq T(n) \leq \sum_{i=0}^{H_2} (8^2/9^2 + 1^2/6^2)^i n^2
$$

Both sides are sum of geometric sequence of common ratio $(8^2/9^2 + 1/6^2) < 1$. Therefore, $T(n) = \Theta(n^2)$.

6. #6 Solution

#6(a)

$$\begin{aligned} T(n) &= T(\log n) + 3 \\ &= T(\log\log n) + 3 + 3 \\ &= T(\log\log\log n) + 3 + 3 + 3 \\ &\cdots \\ &= T(1) + 3H \end{aligned}$$

where H is the height of the recursion tree. Then, $H = \log^* n$, the iterated log of n, which is the number of times the logarithm function must be iteratively applied before the result is less than or equal to 1. Therefore, $T(n) = \Theta(\log^* n)$.

#6(b)

$$\begin{aligned} T(n) &= T(\log n) + 3n \\ &= T(\log\log n) + 3n + 3\log n \\ &= T(\log\log\log n) + 3n + 3\log n + 3\log\log n \\ &\cdots \\ &= T(1) + 3n + 3\log n + 3\log\log n + \cdots + 3\log^* n \\ &< T(1) + 3n + 3\log n + 3\log n + \cdots + 3\log n = \Theta(3n + \log^* n \log n) = \Theta(n) \end{aligned}$$

Therefore, $T(n) = O(n)$. Also, we have

$$T(n) > T(1)+3n+3\log^* n+3\log^* n+\cdots+3\log^* n = \Theta(3n+\log^* n \log^* n) = \Theta(n)$$

which implies $T(n) = \Omega(n)$. Therefore, we have $T(n) = \Theta(n)$.