Lab #5 Problems
Recurrence Relations and Summations (solve on your own paper)

Solve the following recurrence relations using the iteration technique:

1) $T(n) = 2T\left(\frac{n}{2}\right) + 1$, $T(1) = 1$

- $T(n/2) = \underbrace{2T(n/4) + 1}$ Plug in for $T(n/2)$

$T(n) = 2(2T(n/4) + 1) + 1$

$= 4T(n/4) + 2 + 1 = 4T(n/4) + 3$

- $T(n/4) = \underbrace{2T(n/8) + 1}$ Plug in for $T(n/4)$

$T(n) = 4(2T(n/8) + 1) + 3$

$= 8T(n/8) + 4 + 3 = 8T(n/8) + 7$

... We can now see a pattern, in general:

$T(n) = 2^k T\left(\frac{n}{2^k}\right) + 2^k - 1$ for $k \geq 1$

Since we want to get rid of the $T(\ldots)$'s and we know $T(1) = 1$, let's let $\boxed{\frac{n}{2^k} = 1} \rightarrow n = 2^k \rightarrow k = \log_2 n$.

Now we have $T(n) = 2^{\log_2 n} T(1) + 2^{\log_2 n} - 1$

$\boxed{\star \text{ Log rule: } a^{\log_c b} = b^{\log_c a}}$

$T(n) = n + n - 1$

$\boxed{T(n) = 2n - 1}$

This means there are $2n - 1$ operations for an input size n.
∴ The big-O run-time is $\boxed{O(n)}$

2) $T(n) = T(n-1) + n$, $T(1) = 1$

$$\circ\ T(n-1) = \underbrace{T(n-2) + (n-1)}_{\text{Plug in for } T(n-1)}$$

$$T(n) = T(n-2) + (n-1) + n$$

$$\circ\ T(n-2) = \underbrace{T(n-3) + (n-2)}_{\text{Plug in f } T(n-2)}$$

$$T(n) = T(n-3) + (n-2) + (n-1) + n$$

...

We should start to see a pattern, in general:

$$\underline{T(n) = T(n-k) + (n-k+1) + (n-k+2) + \cdots + (n-1) + n}$$

Since we want to get rid of $T(\ldots)$'s, and we know $T(1) = 1$, $\underline{\text{Let } n-k = 1} \rightarrow k = n-1$

$$T(n) = T(n-(n-1)) + (n-(n-1)+1) + (n-(n-1)+2) + \cdots + (n-1) + n$$

$$= T(1) + 2 + 3 + \cdots + (n-1) + n$$

$$= \underline{1 + 2 + 3 + \cdots + n}$$ Hopefully you should recognize this sequence

$$\boxed{T(n) = \frac{n(n+1)}{2}}$$

So there are $\frac{n(n+1)}{2}$ operations for an inputsize n,

meaning the Big-O run-time is $\boxed{O(n^2)}$

3) $T(n) = T\left(\frac{n}{2}\right) + n$, $T(1) = 1$, Hint: $\sum_{i=0}^{\infty} \frac{n}{2^i} = 2n$ (Just get an approximate solution here.)

$$\circ\ T(n/2) = \underbrace{T(n/4) + n/2}_{\text{Plug in for } T(n/2)}$$

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

$$\circ\ T(n/4) = \underbrace{T(n/8) + n/4}_{\text{Plug in for } T(n/4)}$$

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

...

Hopefully we should see the pattern, in general:

$$T(n) = T\left(\frac{n}{2^k}\right) + \underbrace{\frac{n}{2^{k-1}} + \frac{n}{2^{k-2}} + \cdots + \frac{n}{2^0}}$$

We are given $\sum_{i=0}^{\infty} \frac{n}{2^i} = 2n$,

Since we want an approximate solution Let's say this sequence is $2n$.

Now, we still need to get rid of $T(n/2^k)$, Let $n/2^k = 1$

$$T(n) = T(1) + 2n$$

$$= \boxed{2n+1}$$

So there are $2n+1$ operations for an input size of n.

$\therefore$ The Big-O run-time is $\boxed{O(n)}$

4) $T(n) = 4T\left(\frac{n}{2}\right) + 1$, $T(1) = 1$, Hint: $\sum_{i=0}^{k-1} 4^i = \frac{4^k - 1}{4-1}$

- $T(n/2) = \underbrace{4T(n/4) + 1}_{\text{Plug in for } T(n/2)}$

$$T(n) = 4[4T(n/4) + 1] + 1$$

$$= 4^2 T\left(\frac{n}{2^2}\right) + 4^1 + 1$$

- $T(n/4) = \underbrace{4T(n/8) + 1}$ Plug in for $T(n/4)$

$$T(n) = 4^2[4T(n/8) + 1] + 4^1 + 1$$

$$= 4^3 T\left(\frac{n}{2^3}\right) + 4^2 + 4^1 + 1$$

... Hopefully now we should see a pattern, in general:

$$T(n) = 4^k T\left(\frac{n}{2^k}\right) + 4^{k-1} + 4^{k-2} + \dots + 4 + 4^0$$

Since we are given $\boxed{\sum_{i=0}^{k-1} 4^i = \frac{4^k - 1}{4-1}}$, we have:

$$T(n) = 4^k T\left(\frac{n}{2^k}\right) + \frac{4^k - 1}{3}$$

We also want to destroy $T(\dots)$'s, and we know $T(1) = 1$,

Let $n/2^k = 1 \rightarrow n = 2^k \rightarrow k = \log_2 n$

$$T(n) = 4^{\log_2 n} T(1) + \frac{4^{\log_2 n} - 1}{3} = n^{\log_2 4} + \frac{n^{\log_2 4} - 1}{3}$$

$\boxed{T(n) = \frac{4n^2}{3} - 1/3}$, $\boxed{O(n^2)}$