COLLINS KIBET

Recurrence Homework

1.) Unwinding method:

a) $T(n) = 3T(n-1)$ for $n > 1$; $T(1) = 4$

Recursive case ~~relation~~: $3T(n-1)$

Base case: $T(1) = 4$

Unwinding process

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

$$= 3 \cdot 3\,T(n-2) = 3^2\,T(n-2)$$

$$= 3 \cdot 3 \cdot 3\,T(n-3) = 3^3\,T(n-3)$$

$$\vdots$$

General Formular $\Rightarrow$ $3^k\,T(n-k)$

At base case, $T(1) = 4$. $n - k = 1$, $k = n - 1$

$$T(n) = 3^{n-1}\,T(1)$$

$$= 3^{n-1} \cdot 4$$

Asymptotic Time complexity of $T(n) = 3^{n-1} \cdot 4$ is $\theta(3^n)$ because 3^n is the fastest growing term in the function.

b) $T(n) = T(n-1) + n$ for $n > 0$; $T(0) = 0$.

Recursive case: $T(n-1)$

Base case: $T(0) = 0$

Unwinding process:

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

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

$$\cancel{= (T(n-2-1) + n-2) + n = T(n-3) + 2n - 2}$$

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

$n + n - 1$

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

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

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

-2

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

-3

$$= T(n-4) + 4n - 6$$

-4

$$= T(n-5) + 5n - 10$$

$\vdots \quad \vdots \quad \vdots$

General formular $= T(n-k) + kn - c$

(where c is a constant)

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

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

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

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

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

$$= T(n-4) + n - 3$$

$$T(n-4) = T(n-4-1) + n - 4$$

$$= T(n-5) + n - 4$$

At base case, $T(0) = 0$, $T(n) = \cancel{T(0)} + T(n-k) + kn - c$

at $T(0)$, $n = k$.

So, $T(n) = T(0) + n(n) - c = n^2 - c$.

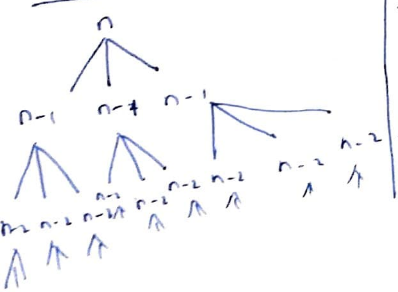
Since n^2 is the fastest growing term, the asymptotic complexity $= \Theta(n^2)$

2.) Recursion tree method

$T(n) = 3T(n-1)$ for $n > 1$; $T(1) = 4$

level	# nodes	work per node	total work
0	$1 = 3^0$	~~n-1~~ 0	
1	$3 = 3^1$	$\cancel{\frac{n-1}{3}}$ ~~3(n-1)~~ 0	
2	$9 = 3^2$	$\cancel{\frac{n-1}{3^2}}$ $3^2(n-1)$ 0	
⋮	⋮		
k	3^k	~~n-1~~ 0	

Recursion Tree



- There's no work done per node. We cannot see this using the recursion tree method.
- Therefore, time complexity of $T(n) = 3T(n-1)$ for $n > 1$; $T(1) = 4$ cannot be analyzed using the recursion tree method.

However, the Unwinding method from part 1 shows that $T(n) = \Theta(3^n)$

3.) Master Method

a) $T(n) = 4T(n/2) + n^2$; $T(1) = 1$.

$a = 4$, $b = 2$, $f(n) = n^2$

We can use the master method to determine the asymptotic bound of $T(n)$.

Comparing $f(n)$ and $n^{\log_b a}$:

$f(n)$	$n^{\log_b a}$	$f(n)$ is asymptotically equal to $n^{\log_b a}$
n^2	$n^{\log_2 4} = n^2$	

$\therefore f(n) = \theta(n^{\log_b a})$

Since $n^2 = n^2$, we apply case 2 from the ~~oth~~ master method procedure:

$\therefore T(n)$ is bounded by $\Theta(n^2 \lg n)$

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

$a = 4$, $b = 2$, $f(n) = n^3$

Compare $f(n)$ and $n^{\log_b a}$:

$f(n) = n^3$; $n^{\log_2 4} = n^2 = \cancel{\Theta}\,\Omega(n^{2+\epsilon})$ | $f(n)$ is polynomially larger than $n^{\log_b a}$

So $f(n) = \Omega(n^{\log_b a + \epsilon})$

Thus, case ~~2~~ 3 applies: ~~$T(n) = \theta(n^2)$~~ ~~$T(n) = \Omega$~~

$T(n) = \theta(n^3) = \theta(f(n))$

Consideration: Is $a * f(n/b) \leq c * f(n)$ for $0 < c < 1$?

$a = 4$

$4 * \frac{n^3}{2^3} \leq c * n^3$ is true. bec

Dividing n^3 by a number on LHS makes it smaller than RHS.

Therefore; $T(n) = \Theta(n^3)$

4.) $T(n) = 3T(n/2) + n\lg n,\ T(1) = 1$

Master method

$a = 3,\quad b = 2,\quad f(n) = n\lg n$

Compare $f(n)$ with $n^{\log_b a}$

$f(n) = n\lg n \quad \Big| \quad n^{\log_b a} = n^{\log_2 3}$

Consider $\log_2 3$. $\log_2 3 > 1$. Using a calculator, $\log_2 3 = 1.585$

Comparing rate of growth of $n\lg n$ and $n^{1.585}$

$n^1 \cdot \lg n < n^{1.585}$

$n^{1.585}$ grows faster than $n^1 \lg n$. Thus,

~~f(n)~~ We consider case 1 in master method procedure

$\cancel{T(n) = \Theta(f(n))}$. $f(n) = O(n^{\log_b a - \epsilon})$

$\cancel{f(n) = \Omega(n^{\log_b a + \epsilon})}$

~~Take~~ $\cancel{n^{1.585+\epsilon}}$

$f(n)$ is polynomially smaller than $n^{\log_b a - \epsilon}$.

We can see that below:

$\dfrac{f(n)}{n\lg n} \qquad \dfrac{n^{\log_b a - \epsilon}}{n^{1.585-\epsilon}} = \dfrac{n^{1.585}}{n^{\epsilon}}$

$n^{\log_b a - \epsilon}$ grows faster than $n\lg n$

Hence, $T(n) = \Theta(n^{\log_2 3})$