Selenay Buse Batibay 30294 CS300

Homework 2

① a

Base case ⇒ $S(1) = 1$ ⇒ when "n" = 1 , the func. prints 1 star

for $n \neq 1$;

↳ if n is even , the func. calls printStarsHelper () and prints 3 stars and calls unnecessaryStars (n/3) recursively

$S(n) = 3S(n/3) + 3$

↳ if n is odd , the func. calls itself recursively with arg. n/3 and calls printStarsHelper

$S(n) = 3S(n/3) + 3$

Therefore ;

$$\left.\begin{array}{ll} S(n) = 1 & \text{if } n = 1 \\ 3S(n) + 3 & \text{if } \underbrace{n \neq 1}_{n \text{ is even either odd}} \end{array}\right\}$$

① b

$$\begin{aligned} S(n) &= 3S(n/3) + 3 \\ &= 3(3S(n/9) + 3) + 3 \\ &= 3^2 S(n/9) + 3^2 . 3 + 3 \\ &= 3^2 . 3 S(n/27) + 3^2 . 3 + 3^2 . 3 + 3 \end{aligned}$$

↳ $S(n) = 3^k S(n/3^k) + 3^{(k-1)} . 4 + 3^{(k-2)} . 4 \dots 3^1 . 4 + 1$

If n is a power of 3 ⇒ $n = 3^k$ ⇒ $\boxed{k = \log_3 n}$

$S(n) = 3^{\log_3 (n)} . S(1) + 4(3^{\log_3 (n) - 1}) + 3^{\log_3 (n) - 2} + \dots + 3^2 + 3 + 1$

$S(n) = n + 2(3^{(\log_3 n)} - 1)$

↳ this means S(n) grows linearly with n

$S(n) \sim O(n)$ //

(2) a

Base case ⇒ $n=0$, $Z(0)=1$

Recursive case ⇒ when $n>0$, the func. calls printZeroHelper(n) that prints $\log(2n)$ zeros. Then calls itself recursively

$$Z(n) = 2Z(n-2) + \log(2n)$$

Therefore,

$$Z(n) = \begin{cases} 1 & \text{if } n=0 \\ 2Z(n-2) + \log(2n) & \text{if } n>0 \end{cases}$$

(2b) $T(n) = 2T(n-1) + \log n$

$T(n-1) = 2T(n-2) + \log(n-1)$ } mult. with 2 both sides

$2T(n-1) = 2^2T(n-2) + 2\log(n-1)$ } mult with k

$2^kT(n-k) = 2^{k+1}T(n-k+1) + 2^k\log(n-k)$

1st step: $T(n) = 2T(n-1) + \log(n)$

2nd ": $2T(n-1) = 2^2T(n-2) + 2^1\log(n-1)$

3rd step: $2^2T(n-2) = 2^3T(n-3) + 2^2\log(n-2)$

} we have cancellations

$T(n) - \sum_{i=2}^{n} 2^i \cdot \log(n-i) \Rightarrow T(n) \leq$ } $T(n) \sim O(2^n \log(n))$