Jordan Allard

Part 1: Set up recurrences, use the Master Theorem

Consider the following pseudo code sketches. For each, state its recurrence, using $T(n)$ to denote the maximum number of steps the function makes on an input of size n. In particular, (1) next to the pseudo code above, highlight its non-recursive parts and estimate their running time in big-Oh notation, and (2) next to the pseudo code above, highlight the recursive parts and estimate their running times using the function $T()$. Then, evaluate it using the Master Theorem.

(a) Function F (A – an array of size n):

    if $n = 1$: return $A[1]$
    use $O(n)$ steps and compute arrays B, C, and D of size $\lfloor n/3 \rfloor$ each
    $x = F(B)$
    $y = F(C)$
    $z = F(D)$
    return $x + y + z$

(i) Set up a recurrence for this pseudo code sketch. Do not worry about the rounding. Do not forget the base case.

$T(n) = 3T(n/3) + O(n) = 3(3T(n/9) + c\cdot\frac{n}{3}) + cn = 9T(n/9) + 2cn$

$T(n) \leq 3^k T(n/3^k) + kcn$

$$T(n) = \begin{cases} O(1) & \text{if } n=1 \\ 3T(n/3) + O(n) & \text{if } n>1 \end{cases}$$

(ii) Analyze the recurrence using the Master Theorem.

(b) Function G (A – an array of size n):

if $n = 1$: return $A[1]$

use $O(n)$ steps and compute arrays B and C of size $\lfloor n/3 \rfloor$ each

$x = F(B)$

$y = F(C)$

return $x - y$

(i) Set up a recurrence for this pseudo code sketch. Do not worry about the rounding. Do not forget the base case.

$$T(n) = \begin{cases} O(1) & \text{if } n=1 \\ 2T(n/3) + O(n) & \text{if } n>1 \end{cases}$$

(ii) Analyze the recurrence using the Master Theorem.

(c) Function H (A – an array of size n):

if $n = 1$: return $A[1]$

use $O(n)$ steps and compute arrays B, C, D, and E of size $\lfloor n/3 \rfloor$ each

$x = H(B)$

$y = H(C)$

$z = H(D)$

$w = H(E)$

return $x + y + z + w$

(i) Set up a recurrence for this pseudo code sketch. Do not worry about the rounding. Do not forget the base case.

$$T(n) = \begin{cases} O(1) & \text{if } n=1 \\ 4T(n/3) + O(n) & \text{if } n>1 \end{cases}$$

(ii) Analyze the recurrence using the Master Theorem.