CMPS 2200 Assignment 2

Zoe Murphy

Part 1. Asymptotic Analysis

Derive Asymptotic Upper Bounds of Work

1) $W(n) = 2W(\frac{n}{3}) + 1$ ← Doesn't grow, constant

$a = 2$

$b = 3$ $\quad \log_3 2 >$

$f(n) = 1$ $\quad n^{0.6}$ will grow

$\boxed{O(n^{\log_3 2})}$

2) $W(n) = 5W(\frac{n}{4}) + n$

$a = 5$ $\quad \log_4 5 = 1.16$

$b = 4$

$f(n) = n$ $\quad \underline{\underline{n^{1.16}}}$ vs. n^1

Dominant, so $O(n^{\log_4 5})$

3) $W(n) = 7W(n/7) + n$

$a = 7$ $\quad \log_7 7 = 1$

$b = 7$ $\quad n^{\log_7 7} \approx n$ $\quad$ Balanced

$f(n) = n$

$O(n \log n)$

4) $W(n) = 9W(\frac{n}{3}) + n^2$

$a = 9$ $\quad \log_3 9 = 2$ $\quad$ Balanced

$b = 3$ $\quad n^{\log_3 9} = n^2$

$f(n) = n^2$

$O(n^2 \log n)$

5) $w(n) = 8w(n/2) + n^3$

$a = 8$

$b = 2$ $\qquad \log_2 8 = 3$

$f(n) = n^3$ $\qquad n^3 = n^3$ Balanced

$\boxed{O(n^3 \log n)}$

6) $w(n) = 49w(\frac{n}{25}) + n^{3/2} \log n$ ← Dominant Term

$a = 49$

$b = 25$ $\qquad \log_{25} 49 = 1.21$

$f(n) = n^{3/2} \log n$ $\qquad n^{1.21} < n^{1.5} \log n$

$\boxed{O(n^{3/2} \log n)}$

7) $w(n) = w(n-1) + 2$

$a = 1$ $\qquad w(1) = w(0) + 2$

$b = 1$ $\qquad w(2) = w(1) + 2n$ $\qquad \boxed{O(n)}$

$f(n) = 2$ $\qquad \log_1 1 = 1$ $\quad 1 \quad 2$

constant n time

8) $w(n) = w(n-1) + n^c$, with $c \geq 1$

Dominant

$(n-1)^c$

$(n-2) + (n-1)^c$ $\qquad n-1 < n^c$ $\qquad \boxed{O(n^{c+1})}$

$(n-3) + (n-2)^c$

$w(0) + k^c$

9) $w(n) = w(\sqrt{n}) + 1$

$\sqrt[k]{n} + 1$ $\qquad \boxed{O(\log \log n)}$

$n^{1/2^k}$

$\frac{\log n}{2^k}$

$n = \log^{\log n}$

Part 2 Algorithm Comparison

- Algorithm A solves problems by dividing them into 5 subproblems of half the size, recursively solving each subproblem, & then combining the solutions in linear time
- Algorithm B solves problems of size n by recursively solving 2 subproblems of size n-1 & then combining the solutions in constant time
- Algorithm C solves problems of size n by dividing them into 9 subproblems of size n/3, recursively solving each problem, & then combining solutions in $O(n^2)$ time.

What are the asymptotic running times of each of these algorithms? Which algorithm would you choose?

Algorithm A:

$\underset{\text{\# of subproblems}}{a=5}$ $\quad \underset{\text{half the size}}{b=2}$ $\quad \underset{\text{linear time}}{f(n)=O(n)}$

$$W(n) = 5W(\tfrac{n}{2}) + n$$

$$\log_2 5 = 2.32$$

$$n^{\log_2 5} > n^1$$

$$\boxed{O(n^{\log_2 5})}$$

Algorithm B:

$a=2 \quad b=n-1 \quad \underset{\text{constant time}}{f(n)=1}$

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

root: 1
Level 1: 2
level 2: 4

$$\boxed{O(2^n)}$$

Algorithm C : $a=9$ $b=\frac{n}{3}$ $f(n)=n^2$

$W(n) = 9W(\frac{n}{3}) + n^2$

$\log_3 9 = 2$

$n^2 = n^2$ Balanced

$\boxed{O(n^2 \log n)}$

Choosing an Algorithm:

$$\overset{C}{O(n^2 \log n)} < \overset{A}{O(n^{\log_2 5})} < \overset{B}{O(2^n)}$$

Pick C, it has the best big O time

3) Paranthesis Matching

3a) Coding

3b) Work, Span, & Big O

$W(n) = \underline{W(n-1)} + O(1)$

$\underbrace{W(n) = O(1) + O(1) + \dots}_{n \text{ times}}$

$\boxed{\text{Work } O(n)}$

$S(n) = S(n-1) + O(1)$

Sequential, so recursion depth is $O(n)$

$\boxed{\text{Span } O(n)}$

3c) Code

3d) Recurrences & Span & Work

Work constant time $O(n)+O(\dots$

n is input list

constant time n times

$\boxed{\text{Work is } O(n)}$

Parallel reduction & operations

Tree Depth is $O(\log n)$

$\boxed{\text{Span is } O(\log n)}$

3e) Code

3f) Work & Span & Big O

$2W(\frac{n}{2}) + O(1)$

$(\log_2 2 = 1)$ $O(1)$ $O(1)$

$\boxed{\text{Work} = O(n)}$

Parallel Operations

$\boxed{\text{Span is } O(\log n)}$