CS4310 - Design & Analysis of Algorithms
Analysis
Spring 2017

Mariia Kravtsova

2/7/2017

Problem 1

Algorithm swap(A,n)
Input: Array A containing integer numbers and integer n which shows the length of the array
Output: Array A which its even indexed elements are swapped with their consecutive odd indexed elements

Best and worst case are the same 12n + 3, but the i is increasing by 2 so it is growing faster so 6n+3, so complexity is $O(n)$.

Operation	Counts
i ← 0	1
while (i < n)	n+1
temp ← A[i]	2n
A[i] ← A[i+1]	4n
A[i+1] ← temp	3n
i ← i+2 temp	2n
return A temp	1

Problem 2

Characterize each of the following recurrence equations using the master theorem (assuming that $T(n) = c$ for $n < d$, for constants $c > 0$ and $d \geq 1$). Show all work and give the big-Theta notation of the complexity of each.

For each part, make sure to give the Master Theorem Case that applies and the value(s) of ϵ, δ, and/or k (as applicable).

(a) $T(n) = 2T(n/2) + log(n)$

$case\ 1 : log_2 2 = 1,\ \epsilon = 0.90, \Theta(n^1)$

(b) $T(n) = 8T(n/2) + n^2$

$case\ 1 : log_2 8 = 3,\ \epsilon = 1,\ \Theta(n^3)$

(c) $T(n) = 16T(n/2) + (nlog(n))^4$

$case\ 2 : log_2 16 = 4,\ k = 3,\ \Theta((nlog(n))^5)$

(d) $T(n) = 7T(n/3) + n$

$case\ 1 : log_3 7 = 1.77,\ \epsilon = 1.77,\ O(n^{log_3 7})$

(e) $T(n) = 9T(n/3) + n^3 log(n)$

$case\ 3 : log_3 9 = 2,\ \epsilon = 1, \delta = 0.2,\ O(n^3 log(n))$

Problem 3

Since it recursevly sorts 3 times the 2n/3 elements, we know that a = 3, b = 2/3, and n is the size of our input size. So the best and worst case will be the same. Our c is a constant time of the array-passing which is the base case.
Therefore $T(n) = 3T(2n/3) + c$. So we itterate:
$= 9T(4n/9) + 1 + 3$
$= 1 + 3 + 3^2 + .. + 3^{log_{3/2}n}$
skipping some steps because logs are hard to write and I used wolframAlfa to give me the result.
$= \Theta(n^{1/(log_3 3/2)})$
$= \Theta(n^{2.71})$

Master theorem:
$case\ 1 : log_{2/3}3,\ \epsilon = 0.1, \Theta(n^{log_{2/3}3}),\ O(n^{2.6})$

Problem 4

The only thing changing is the size of our chunks. Ex: n=8 we will see n is going to go down by 2 each time. When n=8 it will go down by 4. The n is now divided into 3/4 each time.
Therefore $T(n) = 3T(3n/4) + c$.

Master theorem:
$case\ 1 : log_{3/4}3,\ \epsilon = 1\ O(n^{2.8}),\ \Theta(n^{log_{3/4}3})$