mir hossain

cs477

hw2

problems

1-5

Mir Hossain

CS 477

HW #2

1.) Recursive Algor.

A) Algorithm finds the minimum value in an array

B.) Recurrance Relation ;

↓ Basic Operation is "temp ≤ A[n-1]

↓ $T(n) = (n-1) + c$

↓ Base Condition => $T(1) = c + c ; T(2) + c + 2c ; \ldots$

↓ Inductive Step : $T(k) = kc$

↓ $T(k+1) = T(k) + c = kc + c = c(k+1) = nc$

↓ $T(n) = nc$ => drop Constant => $O(n)$

2.) A) $T(n) = 2T(n/2) + n^3$

$a = 2; b = 2; c = 3$

↓ $a < b^c \Rightarrow 2 < 2^3$ ✓ => then $O(n^c) \Rightarrow T(n) = O(n^3)$

B) $T(n) = T(\sqrt{n}) + 1 \Rightarrow 2^{k} \cdot T(2k) = T(2^{k/2}) + 1$

↳ $S(k) = S(k/2) + 1 \Rightarrow a = 1, B = 1 \Rightarrow S(k) = \log(a/b) \cdot \log$

$= O(\log k) \Rightarrow k = \log \Rightarrow T(n) = \log(n) \cdot \log$

2 c) Continued

c.) $T(n) = 3T(n/2) + n\log n$

$\Theta(n^{\log}(\log(n)^k) \Rightarrow n^{\log 3/2}(\log(n)^{k+1})$

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

$\Rightarrow$ Dominating Term is $\log(n)^2$

3a) Recursion Tree $T(n) = T(n/4) + T(n/2) + n^2$

$C(n^2)$

$T(n/4)^2 \quad C(n/2)^2 \rightarrow \frac{C(2^2+1)n^2}{4^2} = \frac{5C(n^2)}{4^{2+1}}$

$T(n/16) \quad T(n/8) \quad T(n/8) \quad T(n/4)$

$C(n/4^2)^2 \quad C(n/2^3)^2 \quad C(n/2^3)^2 \quad C(n/2^2)^2 \Rightarrow \frac{n^2+2^3+n^2+2^3n^2+4^2n^2}{4^{2+1+1}}$

$C(n/4^3)^2 \quad C(n/2^5)^2 \quad C(n/2^6)^2 \quad C(n/2^4)^2 \quad C(n/2^5)^2 \quad C(n/2^3)^2 \quad C(n/2^4)^2 \quad C(n/2^3)^2$

$\rightarrow \frac{Cn(1+2^2+2^3+2^4)}{4^{2+1+1+1+1}} \approx \frac{C5^3n^2}{4^6}$; $n = 2^k$; $k = \log_2(n)$

$\rightarrow$ Geometric Sum: $S_n = \frac{r^{n+1}-1}{r-1} \Rightarrow T(n) = Cn^2\frac{(5/16)^{\log_2 n+1}-1}{5/16-1}$

$\Rightarrow n^2 > \underbrace{\frac{(5/16)^{\log(n)+1}-1}{}}_{\text{Constant}} \Rightarrow n^2$ dominating Term

$\Rightarrow O(n^2)$

3.) Continued

Iteration Method:

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

↳ $T(n/2) = 4[4T(\frac{n}{4}) + (n/2)] + n$

↳ $T(n/4) = 4[4[4T(n/8) + (n/4) + n/2) + n$

↳ $n + 2n + 4n + 8N \ldots$ ect

$T(n) = 4^i + \sum_{i=0}^{i-1} 2^i(n)$; $i = \log_2(n)$

↳ $T(n) = n^{\log_2(n)} + \sum_{i=0}^{i-1} 2^i(n)$

$= n^2 + \Sigma 2^i(n)$

$= n^2 + (1+2+4+8+16 \ldots)(n)$

↳ Drop Constant

$n^2 + n \Rightarrow n^2$ dominating Term

$O(n^2)$

4.) Extra Credit

A) Recurrance Relation

If $n = 1$; Return 1

else return $Q(n-1) + 2 * n - 1$

$\Rightarrow$ Finished on next Page

A) $Q(n) = Q(n-1) + 2n - 1 \Rightarrow Q(n-2) + 2n - 2 + 2n - 1 \Rightarrow$

$Q(n-3) + 2n - 4 + 2n - 2 + 2n - 1 \Rightarrow Q(n-4) + 2n - 6 + 2n - 4 + 2n - 2 + 2n$

$Q(1) + 2n(n+1) - n - 2$

$Q(1) = 1$

$Q(2) = Q(2-1) + 2 * 2 - 1 = 4 \Rightarrow$ Functions finds square of a number

$Q(3) = Q(3-1) + 2 * 2 - 1 = 9$

$\hookrightarrow$ ALGORITHM computes square of a number

$Q(n-1) + 2n - 1 = (n-1)^2 + 2n - 1 = n^2 \Rightarrow O(n^2)$

B.) Recurrence Relation for num of multiplications $\rightarrow$

$M(n) = M(n-1) + 1;\ n > 1 \Rightarrow M(1) = 0$

$\hookrightarrow M(n) = n - 1$

5.)

Mystery(n)

$S \leftarrow 0$

for $i \leftarrow 1$ to n do)

$S \leftarrow S + i \times i$

Return S

A) $S \leftarrow S + i \times i$

$S(1) = 1 + 1 = 2$

$S(2) = 2 + 2 \times 2 = 2 + 4 = 6$

↳ This algorithm calculates the sum of Squares of n ints

B.) There's one loop, so complexity is n

5