



CPE 11040830: Algorithms, Homework #2
Dr. Khalil Yousef

CLR 4.1-6 (Second Edition) [Similar to problem 4.3-9 Third Edition with some changes]

Solve the recurrence $T(n) = 2T(\sqrt{n}) + 1$ by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral?

CLR 4.2-4 (Second Edition) [Similar to problem 4.4-8 Third Edition]

Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(n - a) + T(a) + cn$, where $a \geq 1$ and $c > 0$ are constants.

CLR 4.2-5 (Second Edition) [Similar to problem 4.4-9 Third Edition]

Use a recursion tree to give an asymptotically tight solution to the recurrence $T(n) = T(\alpha n) + T((1 - \alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$ and $c > 0$ is also a constant.

CLR 4.3-1 (Second Edition) [Similar to problem 4.5-1 Third Edition with some changes]

Use the master method to give tight asymptotic bounds for the following recurrences.

- a. $T(n) = 4T(n/2) + n$.
- b. $T(n) = 4T(n/2) + n^2$.
- c. $T(n) = 4T(n/2) + n^3$.

CLR 4.3-3 (Second Edition) [Similar to problem 4.5-3 Third Edition]

Use the master method to show that the solution to the binary-search recurrence $T(n) = T(n/2) + \Theta(1)$ is $T(n) = \Theta(\lg n)$.

CLR 4.3-4 (Second Edition) [Similar to problem 4.5-4 Third Edition]

Can the master method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \lg n$? Why or why not? Give an asymptotic upper bound for this recurrence.

CLR 4.4-2 (Second Edition) [Similar to problem 4.6-2 Third Edition]

Show that if $f(n) = \Theta(n^{\lg_b a} \lg^k n)$, where $k \geq 0$, then the master recurrence has solution $T(n) = \Theta(n^{\lg_b a} \lg^{k+1} n)$. For simplicity, confine your analysis to exact powers of b .

CLR 4-1 Recurrence examples (Second Edition) [Similar to problem 4-1 Third Edition with some changes]

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences.

Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible, and justify your

answers.

- a. $T(n) = 2T(n/2) + n^3$.
- b. $T(n) = T(9n/10) + n$.
- c. $T(n) = 16T(n/4) + n^2$.
- d. $T(n) = 7T(n/3) + n^2$.
- e. $T(n) = 7T(n/2) + n^2$.
- f. $T(n) = 2T(n/4) + \sqrt{n}$.
- g. $T(n) = T(n - 1) + n$.
- h. $T(n) = T(\sqrt{n}) + 1$

CLR 4-2 - Finding the missing integer (Second Edition)

An array $A[1 \dots n]$ contains all the integers from 0 to n except one. It would be easy to determine the missing integer in $O(n)$ time by using an auxiliary array $B[0 \dots n]$ to record which numbers appear in A . In this problem, however, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is "fetch the j th bit of $A[i]$," which takes constant time. Show that if we use only this operation, we can still determine the missing integer in $O(n)$ time.

CLR 4-4 - More recurrence examples (Second Edition) [Similar to problem 4-3 Third Edition with some changes]

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for sufficiently small n . Make your bounds as tight as possible, and justify your answers.

- a. $T(n) = 3T(n/2) + n \lg n$.
- b. $T(n) = 5T(n/5) + n/\lg n$.
- c. $T(n) = 4T(n/2) + n^2 \sqrt{n}$
- d. $T(n) = 3T(n/3 + 5) + n/2$.
- e. $T(n) = 2T(n/2) + n/\lg n$.
- f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$.
- g. $T(n) = T(n - 1) + 1/n$.
- h. $T(n) = T(n - 1) + \lg n$.
- i. $T(n) = T(n - 2) + 2 \lg n$.
- j. $T(n) = \sqrt{n}T(\sqrt{n}) + n$.