Recurrences: T(n)=c if n=1; T(n)=3T(n/4)+cn2 otherwise

input size to

each execution

# of

executions

cn2(3/16)(logn–1)

3(logn–1)

c

cn2(1/16)(logn–1)

n/4log4n=n/n=1

log4n

(base case level)

33cn2(1/16)3=cn2(3/16)3

32cn2(1/16)2=cn2(3/16)2

31cn2(1/16)=cn2(3/16)1

c(n/64)2=cn2(1/16)3

c(n/16)2=cn2(1/16)2

32

33

31

3

2

1

30

0

total work

work done by

each execution

Recursion Tree (note: all logarithms are to the base 4)):

level #

n

cn2=cn2(3/16)0

cn2

c(n/4)2=cn2(1/16)

n/4

n/42

n/43

n/4(log4n–1)

(log4n–1)

3log4n=nlog43

cnlog43

Depth of the tree is log of n to the base 4.

We use the property where a and b are constants when calculating the # of executions at the base case level.

Next page shows the Recursion Tree in a tabular format.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| level | Level number | Total # of recursive executions at this level | Input size to each recursive execution | Work done by each recursive execution, excluding the recursive calls | Total work done by the algorithm at this level |
| 0 | 0 | 30 | n | cn2 | cn2(3/16)0 |
| 1 | 1 | 31 | n/4 | cn2(1/16) | cn2(3/16)1 |
| 2 | 2 | 32 | n/42 | cn2(1/16)2 | cn2(3/16)2 |
| Level just above base case level | (log4n–1) | 3(log4n–1) | n/4(log4n–1) | cn2(1/16)(log4n–1) | cn2(3/16)(log4n–1) |
| Base case level | log4n | nlog43 | n/4(log4n) | c | cnlog43 |

ba

Total work done by all recursive executions can be obtained by vertically adding the column "total work."

See that this sum for levels 0 to (log4n–1) can be written as a summation; then add the last work term corresponding to the base case level.

This summation is hard to evaluate, but we can use the result (see Appendix A of text) to evaluate this summation by noting that (3/16)<1 and so