Not true
- $f(n) = n$, when $n/2$, $f(n/2) = n/2$, much smaller than n
- n grows faster than $f(n/2)$

So, true for slow growing funct like "$\log n$"
false for fast growing funct like
"n"
"n^2"
"2^n"

Chapter 4

a) $T(n) = 5T(n/3) + n\lg n$

Master theorem (how fast $T(n)$ grows)

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

1) $\underset{\text{num of recursive calls}}{a = 5}$, $\underset{\text{size of each subproblem is } n/3}{b = 3}$, $f(n) = n\lg n$ extra work outside the recursion

2) Compare $f(n)$ w/ $n^{\log_b a}$

$\log_b a = \log_3 5 = \frac{\log 5}{\log 3} \rightarrow$ growth rate of recursive part $n^{1.46}$

3) Compare $f(n) = n\log n$ w/ $n^{\log_3 5}$

$n\log n$ grows faster than $n^{\log_3 5}$ be $\log n$ adds extra growth factor

4) Since $f(n)$ growth faster $n^{\log_3 5}$

$$T(n) = \Theta(n\lg n)$$

$T(n)$ grows at the same rate as $n\lg n$

Master Theorem, how fast $T(n)$ grows

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$

a - num of smaller problems function split into

b - how much smaller each problem is the input size / by b

f(n) - extra work done outside recursion like sorting, merging data

1. To use compare 2 growth rates:
recursive part which grows like

$n^{\log_b a}$ - means the growth of the recursive tree, shows how much work recursion contributes

2. the non recursive part, which is $f(n)$
this is the extra work done in each step outside recursion

now 3 scenarios:

1. if $f(n)$ grows slower, recursion dominates, and $T(n) \sim n^{\log_b a}$
2. if $f(n)$ grows at the same rate, the growth is $T(n) \sim n^{\log_b a} \cdot \log n$
3. if $f(n)$ grows faster, ~~or~~ $T(n) \sim f(n)$ (the extra work dominates)

b) $T(n) = 3T(n/3) + n/\log n$

$T(n) = aT(\frac{n}{b}) + f(n)$ – recurrence is

1. $a = 3$ – num of recursive calls
2. $b = 3$ – size of each subproblem $\frac{n}{3}$
3. $f(n) = \frac{n}{\log n}$ – the extra work outside the recursion
4. calculate growth rate of recursive part, $n^{\log_b a}$

   $\log_b a = \log_3 3 = 1$ $(\log_3 3 = 1) \rightarrow$

   So, recursion part grows like $n^1 = n$
5. compare $f(n) = \frac{n}{\log n}$ w/ $n^{\log_b a} = n$ (linear)

   compare $f(n)$ w/ n

   $f(n) = \frac{n}{\log n}$ grows slower than n because $\log n$ in the denominator makes $f(n)$ smaller as n gets larger
6. Since $f(n)$ grows slower than $n^{\log_b a} = n$, the recursion dominates

   $T(n) \sim n^{\log_b a} = n^1 = n$

   $T(n) = \Theta(n)$

c) $T(n) = 8T(n/2) + n^3 \sqrt{n}$

$a = 8$ $b = 2$ $f(n) = n^3 \sqrt{n}$

1. ~~calc~~ $n^{\log_b a} = n^{\log_2 8} = 3$ (since $2^3 = 8$)

   So recursive part grows like n^3 (exponent) (polinom)
2. compare $f(n) = n^3 \sqrt{n}$ w/ $n^{\log_b a} = n^3$

$f(n) = n^3\sqrt{n} = n^{3.5}$ (because $\sqrt{n} = n^{\frac{1}{2}}$) $= n^{3.5}$

compare $n^{3.5}$ w/ n^3 (recursive), $n^{3.5}$ grows faster

so, the extra work dominates $(3.5 > 3)$

$$T(n) = \Theta(n^{3.5})$$

---

master theorem determines which part dominates: recursion ($n^{\log_b a}$) or extra work $f(n)$

Regardless of which dominates, final growth rate (the Θ) is based on dominating part

d) $T(n) = 2T(n/2 - 2) + n/2$

1. recursive work creates 2 subproblems $\frac{n}{2} - 2$
2. non recursive work is $\frac{n}{2}$
3. 2nd level of tree
   each $T(\frac{n}{2} - 2)$ is expanded into:
   $T(\frac{n}{2} - 2) = 2T(\frac{n}{4} - 4) + \frac{\frac{n}{2} - 2}{2}$
   now $2 \cdot 2 = 4$ subproblems of size $T(\frac{n}{4} - 4)$
4. non-recursive work
   $2 \cdot \frac{\frac{n}{2} - 2}{2} = \frac{n}{4} - 2$
5. gen pattern at each level
   at depth i, there are 2^i subprob
   size of each is $\frac{n}{2^i} - 2^i$
   none recursive work at depth i is

$T(n) = \Theta(n)$ $\quad 2^i \cdot \frac{\frac{n}{2^i} - 2^i}{2} = \frac{n}{2} - i \cdot 2^i$

e) $T(n) = 2T(n/2) + n/\lg n$

$a = 2$

$b = 2$ $\qquad c = \frac{n}{\log n}$

1. Recursive growth $n^{\log_b a} = n^{\log_2 2} = n^1 = n$
2. extra work $\frac{n}{\log n} = f(n)$
3. compare $\rightarrow$ $\frac{n}{\log n}$ grows slower than n, bc $\log n$ in denom reduces growth of $f(n)$

level 0

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

level 1

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

total extra work is : $\frac{\frac{n}{2}}{\log\frac{n}{2}} + \frac{\frac{n}{2}}{\log\frac{n}{2}} \frac{n}{\log\frac{n}{2}}$

level 2

$T(n/4) = 2T(n/8) + \frac{\frac{n}{4}}{\log n/4}$

tot extra work is

$\frac{\frac{n}{4}}{\log\frac{n}{4}} \cdot 4 = \frac{n}{\log n/4}$, extra work in depth $\frac{n}{\log(n/2^i)}$

Total work $= \sum_{i=0}^{\log_2 n} \frac{n}{\log n/2^i}$

$T(n) = \Theta(n \log\log n)$

f) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

1. there are 3 recursive subproblems w/ size $\frac{n}{2}, \frac{n}{4}, \frac{n}{8}$

2. extra work outside recursion is n

level 0

$$n + T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right)$$

level 1

$$\frac{n}{2} + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + T\left(\frac{n}{16}\right)$$

at depth i, the subproblem sizes are $\frac{n}{2^i}, \frac{n}{2^{i+1}}, \frac{n}{2^{i+2}}$

1. work at each level dominated by sum of all n-values at that level
2. there are more subproblems but at each level they shrink
3. tot work per level adds up to n

$$T(n) = \Theta(n \log n)$$

g) $T(n) = T(n-1) + \frac{1}{n}$

recursive work - $T(n-1)$

extra work $\frac{1}{n}$

$= T(n-1) = T(n-2) + \frac{1}{n-1}$

$= T(n-2) = T(n-3) + \frac{1}{n-2}$

$T(n) = T(n-2) + \frac{1}{n} + \frac{1}{n-1}$

$T(n) = T(n-3) + \frac{1}{n} + \frac{1}{n-1} + \frac{1}{n-2}$

$T(n) = T(1) + \sum_{k=2}^{n} \frac{1}{k} \approx \log n$

Total work: $T(n) = T(1) + \log n$
Since $T(1)$ is constant, dominant term is $\log n$

$T(n) = \Theta(\log n)$

h) $T(n) = T(n-1) + \lg n$

1. Recursive work $T(n-1)$ work done on smaller problem of size $n-1$
2. extra work $\log n$ - addition work outside the recursion

$T(n) = T(n-1) + \log n$
$= T(n-1) = T(n-2) + \log(n-1)$
$= T(n-2) = T(n-3) + \log(n-2)$
Sub expansion back in orig equat
$T(n) = T(n-2) + \log \cancel{n-1} + \log(n-1)$
$T(n) = T(n-3) + \log n + \log(n-1) + \log(n-2)$

expansion will look like: if we continue
$T(n) = T(1) + \sum_{k=2}^{n} \log k$

$\sum_{k=2}^{n} \log k = \log(2 \cdot 3 \cdot 4 \ldots n) = \log(n!)$ $\approx n \log n - n$

Total work: $T(n) = \Theta(n \log n)$

i)* $T(n) = T(n-2) + 1/\log n$
recursive work $T(n-2)$ size
extra work $\frac{1}{\log n}$

$T(n) = T(n-2) + \frac{1}{\log n}$
$T(n-2) = T(n-4) + \frac{1}{\log(n-2)}$
$T(n-4) = T(n-6) + \frac{1}{\log(n-4)}$

sub expansion block

$$T(n) = T(n-4) + \frac{1}{\log n} + \frac{1}{\log(n-2)}$$

$$T(n) = T(n-6) + \frac{1}{\log n} + \frac{1}{\log(n-2)} + \frac{1}{\log(n-4)}$$

...

$$T(n) = T(\text{small number}) + \sum_{k=0}^{\lfloor n/2 \rfloor - 1} \frac{1}{\log(n-2k)}$$

$$\approx \frac{n/2}{\log n}$$

tot work: $T(n) = \Theta\left(\frac{n}{\log n}\right)$

j) $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$

recursive $\sqrt{n}$

extra work n

level 0 $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$ tot work

level 1 $T(\sqrt{n}) = \sqrt{\sqrt{n}}\, T(\sqrt{\sqrt{n}}) + \sqrt{n}$

tot work

$\sqrt{n} \cdot \sqrt{\sqrt{n}}\, T(\sqrt{\sqrt{n}}) + \sqrt{n} \cdot \sqrt{n} = n + \sqrt{n} \cdot \sqrt{n}$

num of subproof $= (\sqrt{n})^{i} = n^{\frac{1}{2} i}$

work done at level i dominated by n

as $\sqrt{n}\, T(\sqrt{n})$ shrinks

$T(n) = \Theta\, n \cdot \log\log n$