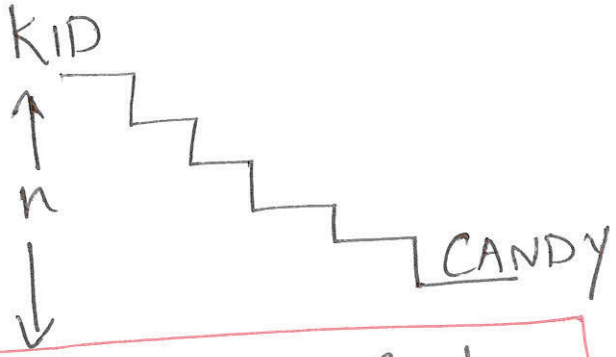
$$T(n) = T(n-1) + C$$
$$T(1) = 1$$

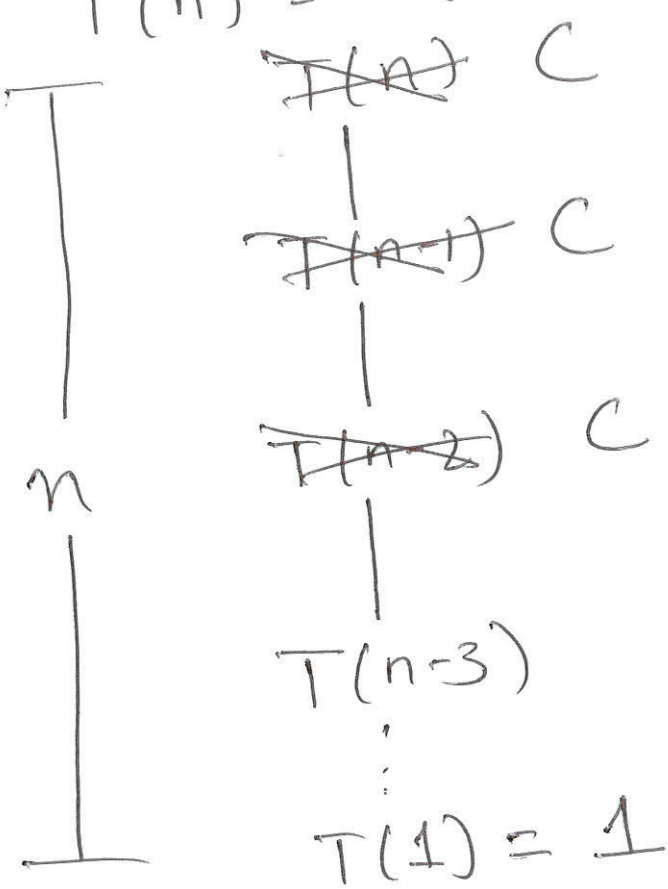


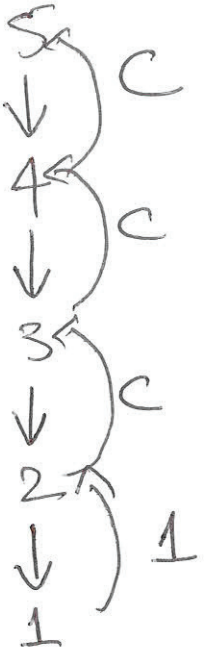
KID getting Candy

$$fib(n) = n * fib(n-1)$$

Finding Fibonacci of n

$$T(n) = T(n-1) + C$$





Total Work $= \underbrace{(C + C + C + C + \cdots)}_{n-1} n-1 + 1$

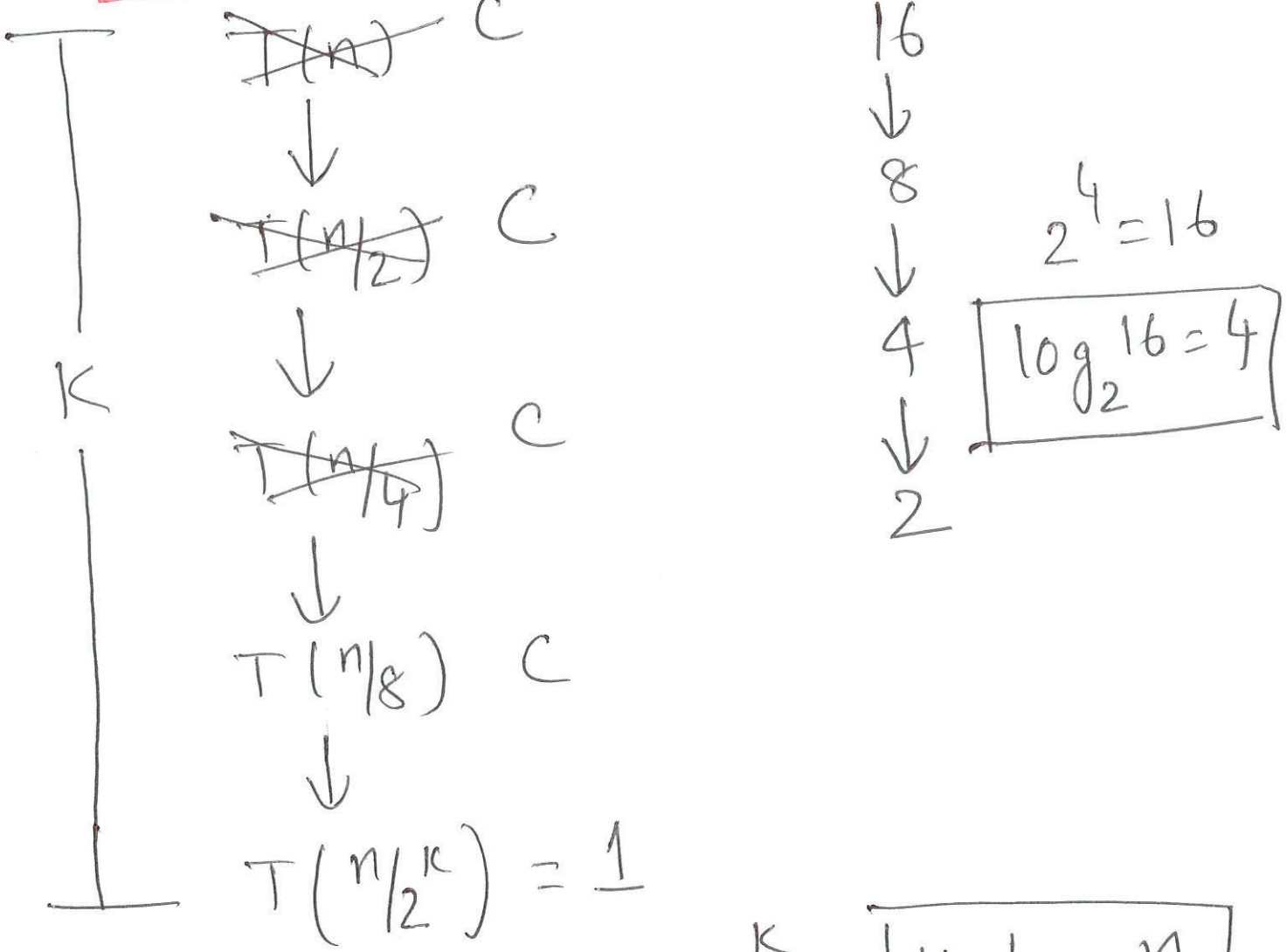
$= C(n-1) + 1$

$= Cn - C + 1 = \Theta(n)$

$= Cn$

$$T(n) = T(n/2) + c$$
$$T(1) = 1$$

Finding A Word in Dictionary



$$\frac{n}{2^k} = 1 \quad \therefore \quad n = 2^k \quad \therefore \boxed{k = \log_2 n}$$

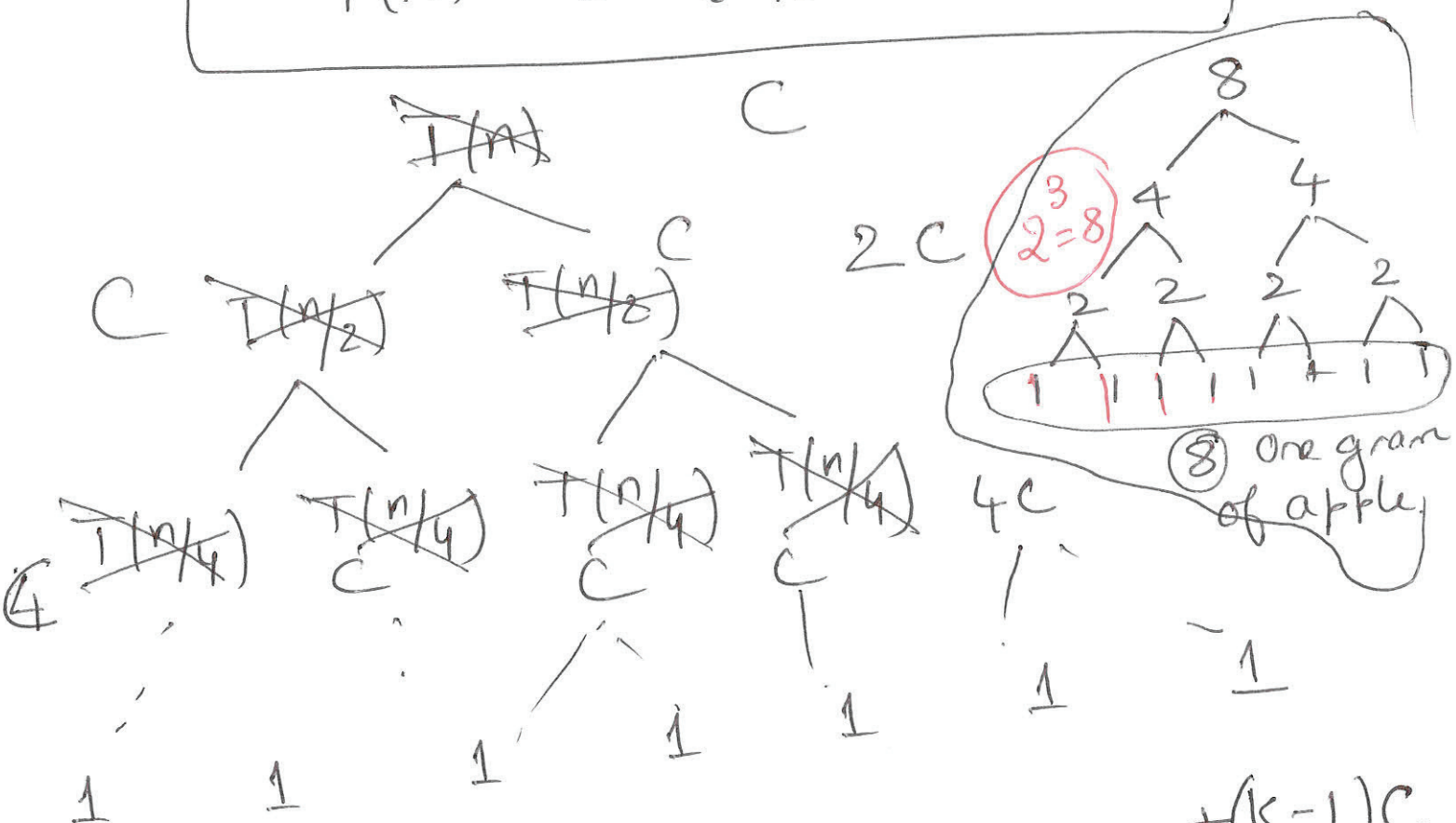
Total work $= k \cdot c$

$= \log_2 n \cdot c = \Theta(\log_2 n)$

$$T(n) = 2T(n/2) + C$$
$$T(1) = 1$$

n grams of Apple.
Cut apple by Half and eat

$$T(n) = T(n/2) + T(n/2) + C$$
$$T(n) = 2T(n/2) + C$$



Total work $T(n) = C + 2C + 4C + 8C + \cdots + (k-1)C$

$$T(n) = C\left(2^0 + 2^1 + 2^2 + \cdots + 2^{k-1}\right)$$
$$= C\left(2^k + 1\right) = 2^k C$$

But $2^k = n$

$$T(n) = C(n+1)$$
$$= Cn = \Theta(n)$$

$$T(n) = T(n/2) + n$$
$$T(1) = 1$$

④

$T(n)$	n	$n/2^0$
$T(n/2)$	$n/2$	$n/2^1$
$T(n/4)$	$n/4$	$n/2^2$
$\vdots$		
$T(1) =$	$n/2^k = 1$	

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$k = \log_2 n$$

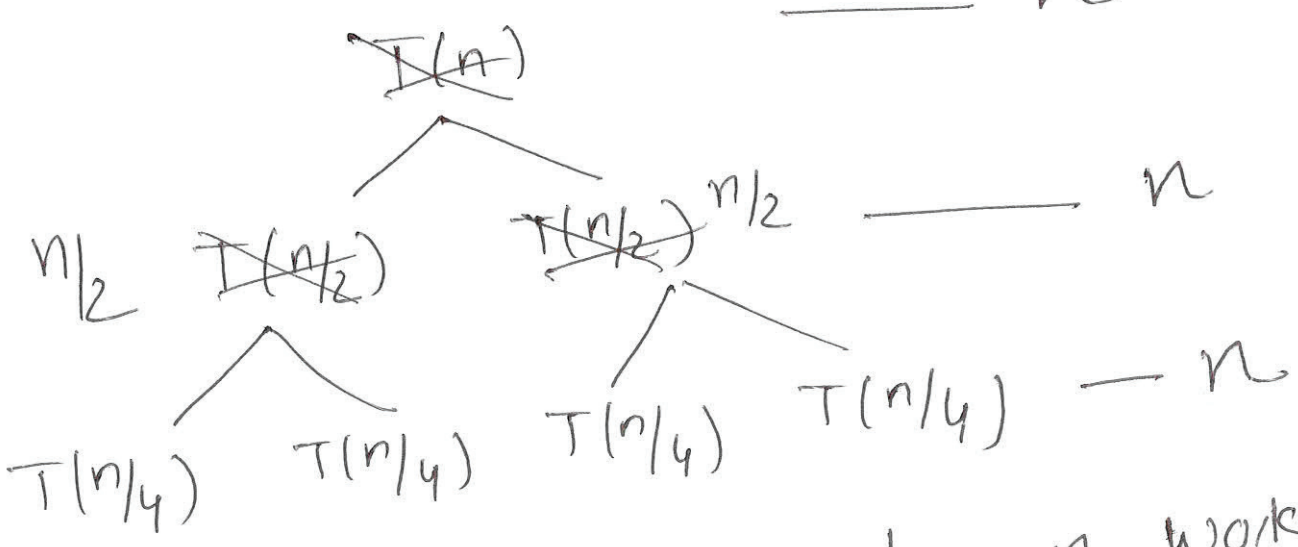
$$\text{Total work} = n\left(1/2^0 + 1/2^1 + 1/2^2 + 1/2^3 + \cdots 1/2^k\right)$$

$$= n(1 + \underbrace{0.5 + 0.25 + \cdots}_{1} \quad 1/2 + 1/3 + 1/4 + \cdots + 1/n = 1$$

$$n(2) = \Theta(n) \qquad T(n) = \Theta(n)$$

$$T(n) = 2T(n/2) + n$$
$$T(1) = 1$$

MERGE SORT



In every level we are doing n work

$$T(n/2^k) = 1 \qquad \frac{n}{2^k} = 1$$

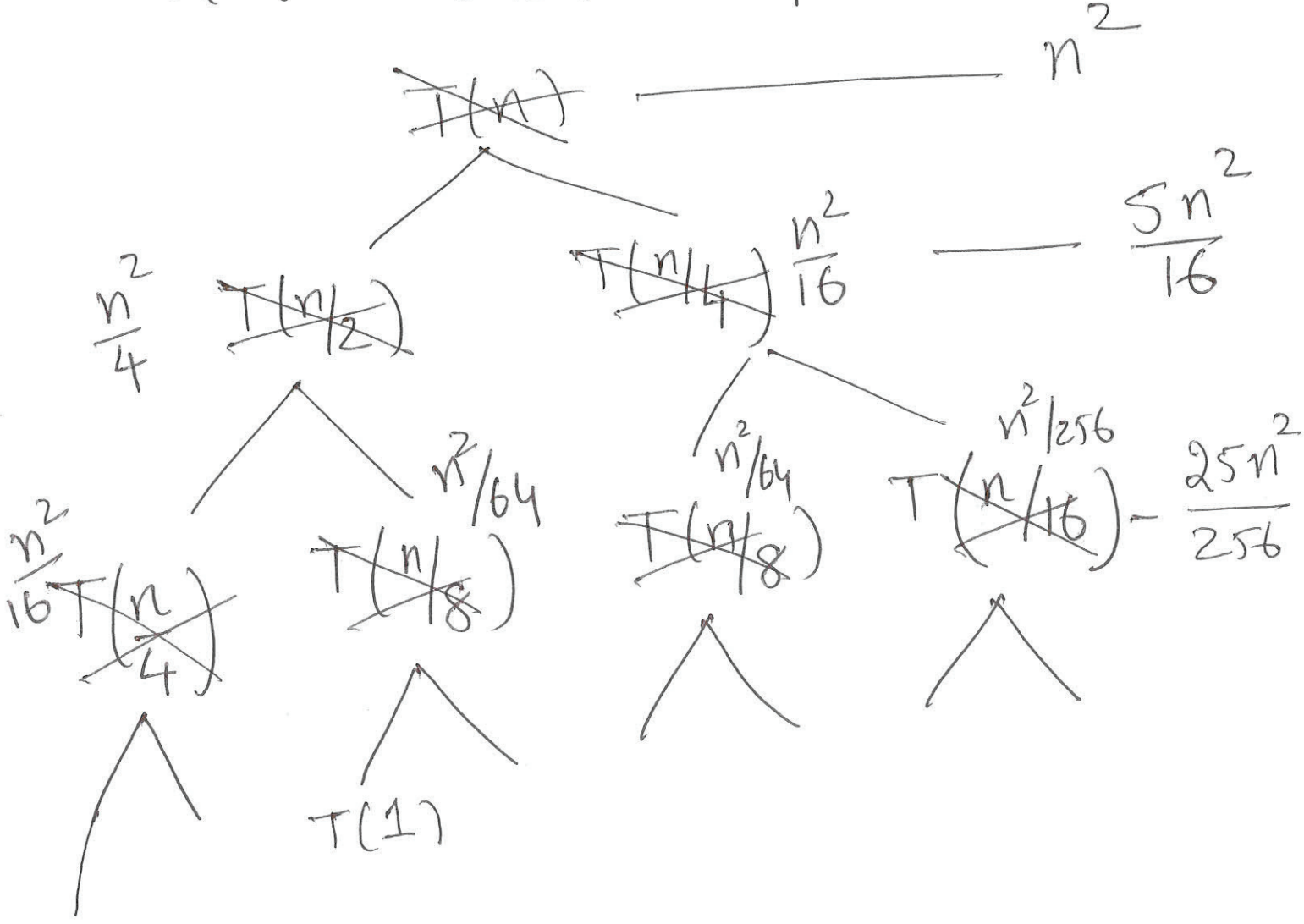
$$k = \log_2 n$$

Tree has $\log_2 n$ Levels.

So total work done $T(n) = n \cdot \log_2 n$

$$T(n) = \Theta(n \log_2 n)$$

$$T(n) = T(n/2) + T(n/4) + n^2$$



How MANY Leaves are there?

This is not so obvious as we are dealing with different speeds

We are solving n problem as $\frac{n}{2} + \frac{n}{4} = \frac{3n}{4}$ Problem

So Number of leaves must be less than n

$$n^2 + \frac{5n^2}{16} + \frac{25n^2}{256} + \cdots + \frac{5^k n^2}{16^k}$$

$$\leq n^2\left(1 + 5/16 + 5^2/16^2 + \frac{5^3}{16^3} + \cdots \frac{5^k}{16^k}\right) \leq 2n^2$$

$$= \Theta(n^2)$$

Smaller than $(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots = 2$

$$T(n) = T(n-1) + T(n-2) + C$$

$$T(1) = 1$$

$$T(0) = 0$$

$$fib(n) = fib(n-1) + fib(n-2)$$

$$fib(1) = 1$$

$$fib(0) = 0$$

$$T(n) = \Theta(1.6^n)$$

$$T(n) = T(n-1) + T(n-1) + C$$

$$T(1) = 1$$

```
Th( int n, int S, int t, int d){
   if(n) {
      Th(n-1, S, d, t)
      S → d
   3  3  Th(n-1, t, S  d)
```

$$T(n) = \Theta(2^n)$$

3.11 Master Theorem

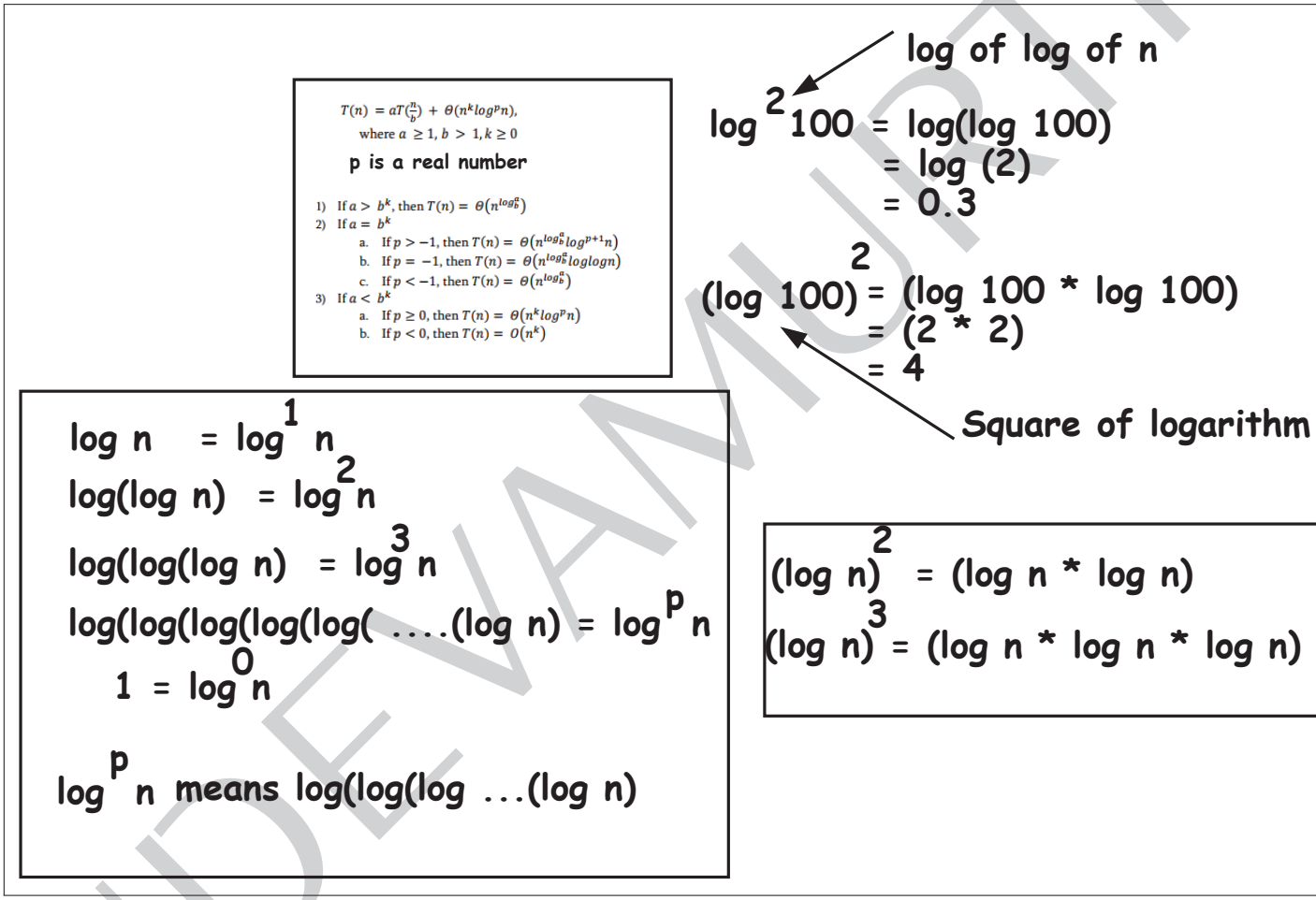


Figure 3.17: Master Theorem

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} log log n)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

1. $T(n) = 3T(n/2) + n^2$
2. $T(n) = 7T(n/2) + n^2$
3. $T(n) = 4T(n/2) + n^2$
4. $T(n) = 3T(n/4) + n \log n$
5. $T(n) = 4T(n/2) + \log n$
6. $T(n) = T(n-1) + n$
7. $T(n) = 4T(n/2) + n^2 \log n$
8. $T(n) = 5T(n/2) + n^2 \log n$
9. $T(n) = 3T(n/3) + n/\log n$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

10) $T(n) = 2T(n/4) + C$

11) $T(n) = T(n/4) + \log n$

12) $T(n) = T(n/2) + T(n/4) + n^2$

13) $T(n) = 2T(n/4) + \log n$

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

15) $T(n) = 8T((n - \sqrt{n})/4) + n^2$

16) $T(n) = 2T(n/4) + \sqrt{n}$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

17) $T(n) = 2T(n/4) + n^{0.51}$

18) $T(n) = 16T(n/4) + n!$

19) $T(n) = 3T(n/2) + n$

20) $T(n) = 4T(n/2) + cn$

21) $T(n) = 3T(n/3) + n/2$

22) $T(n) = 4T(n/2) + n/\log n$

23) $T(n) = 7T(n/3) + n^2$

24) $T(n) = 8T(n/3) + 2^n$

25) $T(n) = 16T(n/4) + n$

$T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \Theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \Theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

$a = 3$ $\quad b = 2$ $\quad k = 2 \quad p = 0$

a	b^k
3	2^2

$3 < 4$

Case 3

$p = 0$

Case 3a

$\Theta(n^2 \log^0 n)$

$= \Theta(n^2)$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 7\,T(n/2) + n^2$

$(a = 7)$ $\boxed{b = 2}$ $\boxed{k = 2 \quad p = 0}$

a	b^k
7	2^2

$7 > 4$

Case 1

$$\boxed{T(n) = \theta(n^{\log_2 7})}$$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 4T(n/2) + n^2$

$a = 4$, $b = 2$, $k = 2$, $p = 0$

$a \quad b^k$

$4 \quad 2^2$

$4 = 4$

Case 2

$p = 0$

Case 2a

$T(n) = \theta(n^{log_2^4} log^{(0+1)} n)$

$T(n) = \theta(n^2 \log n)$

$\log_2^4 = \log_2^{2^2} = 2 \log_2^2 = 2$

4

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 3T(n/4) + n \log n$

$a = 3$, $b = 4$, $k = 1$, $p = 1$

$n^1 \log^1 n$

a	b^k
3	4^1

$3 < 4$

Case 3a

$p = 1$

$T(n) = \theta(n^1 \log^1 n)$

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

~~Case 3~~

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 4T(n/2) + \log n$

$n^0 \log^1 n$

$a = 4$ $b = 2$ $k = 0$ $p = 1$

$a \quad b^k$

$4 \quad 2^0$

$4 > 1$

Caso 1

$T(n) = \theta(n^{\log_2 4})$

$T(n) = \theta(n^2)$

$\log_2 4 = \log_2 2^2 = 2 \log_2 2 = 2$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = T(n-1) + n$

CANNOT BE SOLVED BY MASTER THEORE

$T(n)$ n

$T(n-1)$ $(n-1)$

$T(n-2)$ $(n-2)$

$T(1)$ 1

Total Work $= 1 + 2 + 3 + \ldots + n$

$= \frac{n(n+1)}{2} = \Theta(n^2)$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 4T(n/2) + n^2 \log n$

$a = 4$ $\quad b = 2$ $\quad k = 2$ $\quad p = 1$

$a \quad b^k$

$4 \quad 2^2 \quad p = 1$

$4 = 4$ Case 2a

Case 2

$T(n) = \theta(n^{\log_2 4} \log^{1+1} n)$

$T(n) = \theta(n^2 \log^2 n)$

$\log_2 4 = \log_2 2^2 = 2 \log_2 2 = 2$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 5T(n/2) + n^2 \log n$

$a = 5$ $\quad b = 2$ $\quad k = 2$ $\quad p = 1$

$a \quad b^k$

$5 \quad 2^2$

$5 > 4$

Case 1

$$T(n) = \Theta(n^{\log_2 5})$$

(9)

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

$a = 3$ $\quad$ $b = 3$ $\quad$ CANNOT APPLY MASTER THEOREM

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 2T(n/4) + C$

$n^0 \log^0 n$

$a = 2$ $\quad b = 4$ $\quad k = 0$ $\quad p = 0$

$a \quad b^k$

$2 \quad 4^0$

$2 > 1$

Case 1

$$T(n) = \theta(n^{\log_4 2})$$

$$\boxed{T(n) = \theta(n^{1/2})}$$

$$\log_4 2 = \frac{\log_2 2}{\log_2 4} = \frac{1}{2\log_2 2} = 1/2$$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = T(n/4) + \log n$

$n^0 \log^1 n$

$a = 1$ $\quad b = 4$ $\quad k = 0$ $\quad p = 1$

$a \quad b^k$

$1 \quad 4^0 \quad p = 1$

$1 = 1$ Case 2a

Case 2

$T(n) = \theta(n^{\log_4 1} \log^{1+1} n)$

$= \theta(n^0 \log^2 n)$

$T(n) = \theta(\log^2 n)$

$\log_{\text{any base}} 1 = 0$ $\quad 4^0 = 1$

$\log_4 1 = 0$

(12)

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = T(n/2) + T(n/4) + n^2$

Cannot be solved by master meth

$T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \Theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \Theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 2T(n/4) + \log n$

$n^0 \log^1 n$

$a = 2$ $\quad b = 4$ $\quad k = 0$ $\quad p = 1$

a	b^k
2	4^0
2	1
$2 > 1$	

Case 1

$T(n) = \Theta(n^{\log_4 2})$

$T(n) = \Theta(n^{1/2})$

$\log_4 2 = \frac{\log_2 2}{\log_2 4} = \frac{1}{2}$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

$a = 3$ $\quad b = 3$ $\quad k = 1$ $\quad p = 1$

a	b^k
3	3^1
$3 = 3$	

Case 2

Case 2a $\quad p = 1$

$T(n) = \theta(n^{\log_3^3} \log^{1+1} n)$

$T(n) = \theta(n \log^2 n)$

(15)

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 8T((n-\sqrt{n})/4) + n^2$

MT cannot be used

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,
where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

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

$n^{1/2} \log^0 n$

$a = 2$ $b = 4$

$k = 1/2$ $p = 0$

$a \quad b^k$

$2 \quad 4^{1/2}$ $p = 0$ $T(n) = \theta(n^{\log_4 2} \log^{0+1} n)$

$2 = 2$

Case 2

Case 2 $\Rightarrow$ $T(n) = \theta(n^{1/2} \log n)$

$\log_4^2 = \frac{\log_2^2}{\log_2^4} = \frac{1}{2}$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 2T(n/4) + n^{0.5}\,1$

$a = 2$ $\quad b = 4$ $\quad k = 0.51$ $\quad p = 0$

$a \quad b^k$

0.51

$2 \quad 4$

$2 \quad 2.02$

$2 \quad 4 \quad 2.02$

Case 3

$p = 0$

Case 3a

$T(n) = \theta(n^{0.51} \log^0 n)$

$= \theta(n^{0.51})$

(18)

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 16\,T(n/4) + n!$

$a = 16$ $\quad$ $b = 4$

$= \Theta(n!)$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 3T(n/2) + n$

$n^1 \log^0 n$

$a = 3$ $\quad b = 2$ $\quad k = 1$ $\quad p = 0$

a	b^k
3	2^1
3	2

$3 > 2$

Case 1

$$T(n) = \theta(n^{\log_2 3})$$

$T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \Theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \Theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 4T(n/2) + cn$

$n^1 \log^0 n$

$a = 4$ $b = 2$ $k = 1$ $p = 0$

a	b^k
4	2^1

$4 > 2$

Caso 1

$T(n) = \Theta(n^{\log_2 4})$

$T(n) = \Theta(n^2)$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,
where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 3T(n/3) + n/2$

$a=3$ $b=3$ $\frac{1}{2} n^1 \log^0 n$

$k=1$ $p=0$

a	b^k	
3	3^1	$p=0$
$3=3$		Case 2a

Case 2

$T(n) = \theta(n^{\log_3 3} \log^{0+1} n)$

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

$T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \Theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \Theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 7T(n/3) + n^2$

$a = 7$ $\quad b = 3$ $\quad k = 2$ $\quad p = 0$

$a \quad b^k$

$7 \quad 3^2$

$7 > 6$

Caso 1

$$T(n) = \Theta(n^{\log_3 7})$$

$$\approx \Theta(n^{\log_3 9})$$

$$\boxed{T(n) \approx \Theta(n^2)}$$

$\log_3 9 = \frac{\log_7 7}{\log_7 3} = \frac{1}{\log_7 3}$ $\quad \log_3 3^2 = 2$

$T(n) = aT(\frac{n}{b}) + \Theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \Theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \Theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \Theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \Theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \Theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 8T(n/3) + 2^n$

$T(n) = \Theta(2^n)$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

$T(n) = 16T(n/4) + n$

$a = 16$ $\quad b = 4$ $\quad k = 1$ $\quad p = 0$

a	b^k
16	4^1

$16 > 4$

Case 1

$T(n) = \theta(n^{\log_4 16})$

$= \theta(n^2)$

3.12 Problem set

Problem 3.12.1.

Implement **computing changes for US coins** as explained in figure 3.18

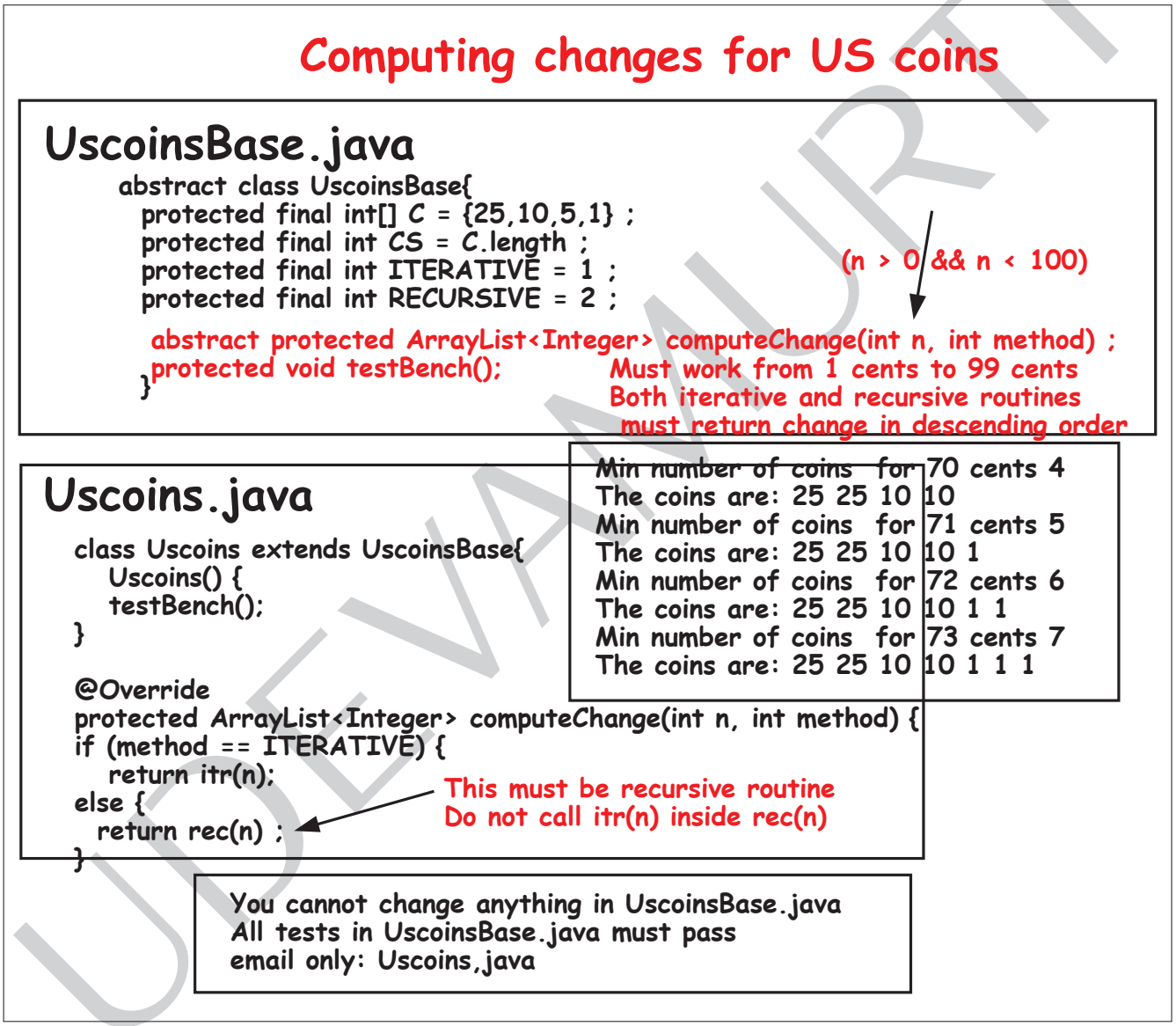


Figure 3.18: Computing changes for US coins

Problem 3.12.2.

Draw on a paper the stack trace of **Tower of Hanoi** problem for 3 disks. Show clealy your final solution. How many steps are required? No code is required.

Problem 3.12.3. Write a procedure called **length** as shown in the figure 3.19. You must use the code given in 3.20 Show on paper, how your code works. Note that you cannot use any loop statements and array length is not known.

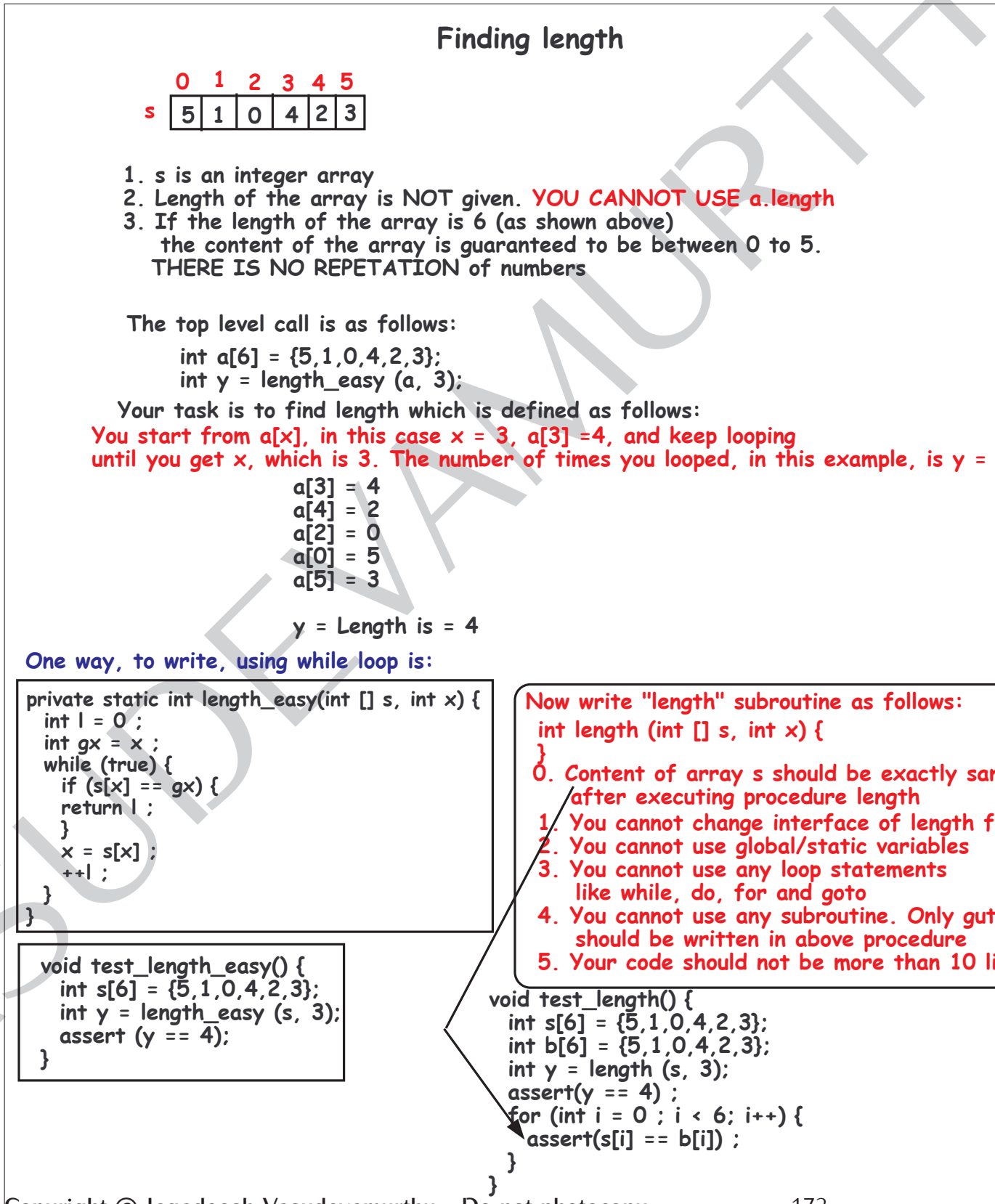


Figure 3.19: Finding the length

Length.java

```
class Length {
  private static final IntUtil u = new IntUtil();

 private static int length_easy(int [] s, int x) {
   int l = 0 ;
   int gx = x ;
   while (true) {
     if (s[x] == gx) {
       return l ;
     }
     x = s[x] ;
     ++l ;
   }
 }

  private static int length(int [] s, int x) {

    //WRITE YOUR CODE ONLY HERE


  }

 public static void testbed() {
    int s[] = {5,1,0,4,2,3} ;
    int y = length_easy(s,3) ;
    System.out.println("length_easy y = " + y);
    u.myassert(y == 4) ;
    int b[] = {5,1,0,4,2,3} ;
    int x = length(s,3) ;
    System.out.println("length x = " + x);
    u.myassert(x == y) ;
    for (int i = 0; i < s.length; ++i) {
      u.myassert(s[i] == b[i]);
    }
    System.out.println("Assert passed");
 }

public static void testbed1() {
   int s[] = {5,1,0,4,2,3} ;
   int b[] = {5,1,0,4,2,3} ;
   int l = s.length ;
   for (int j = 0; j < l ; ++j) {
     int y = length_easy(s,j) ;
     System.out.println("length_easy y = " + y);
     int x = length(s,j) ;
     System.out.println("length x = " + x);
     u.myassert(x == y) ;
     for (int i = 0; i < s.length; ++i) {
       u.myassert(s[i] == b[i]);
     }
     System.out.println("Assert passed");
   }
}

public static void main(String[] args) {
    System.out.println("Length.java");
    testbed() ;
    testbed1() ;
  }
```

Figure 3.20: Write code only in the procedure **length**

Problem 3.12.4. Write a procedure called **fibS that does not uses recursion** as shown in the figure 3.21. You must use the code given in 3.21

```
private static int fibS(int n) {
       setKount() ;
    //WRITE YOUR CODE HERE

  System.out.println("fib of " + n + ": = " + a + " max Stack used (num executed)= " + getKount()) ;
}
```

```
for (int i = 2; i < 30; ++i) { //Run with 300 to see hangs at 50
   int a = fibS(i) ;
}
```

```
fib of 2: = 1 max Stack used (num executed)= 2
fib of 3: = 2 max Stack used (num executed)= 2
fib of 4: = 3 max Stack used (num executed)= 3
fib of 5: = 5 max Stack used (num executed)= 3
fib of 6: = 8 max Stack used (num executed)= 4
fib of 7: = 13 max Stack used (num executed)= 4
fib of 8: = 21 max Stack used (num executed)= 5
fib of 9: = 34 max Stack used (num executed)= 5
fib of 10: = 55 max Stack used (num executed)= 6
fib of 11: = 89 max Stack used (num executed)= 6
fib of 12: = 144 max Stack used (num executed)= 7
fib of 13: = 233 max Stack used (num executed)= 7
fib of 14: = 377 max Stack used (num executed)= 8
fib of 15: = 610 max Stack used (num executed)= 8
fib of 16: = 987 max Stack used (num executed)= 9
fib of 17: = 1597 max Stack used (num executed)= 9
fib of 18: = 2584 max Stack used (num executed)= 10
fib of 19: = 4181 max Stack used (num executed)= 10
fib of 20: = 6765 max Stack used (num executed)= 11
fib of 21: = 10946 max Stack used (num executed)= 11
fib of 22: = 17711 max Stack used (num executed)= 12
fib of 23: = 28657 max Stack used (num executed)= 12
fib of 24: = 46368 max Stack used (num executed)= 13
fib of 25: = 75025 max Stack used (num executed)= 13
fib of 26: = 121393 max Stack used (num executed)= 14
fib of 27: = 196418 max Stack used (num executed)= 14
fib of 28: = 317811 max Stack used (num executed)= 15
fib of 29: = 514229 max Stack used (num executed)= 15
```

Figure 3.21: Fibonacci number using stack

Problem 3.12.5. Write a procedure called **th that does not uses recursion** as shown in the figure 3.22. You must use the code given in 3.22


```
public class TowerOfHanoi {
  private int w = 0 ;
  public void resetKount() {w = 0; }
  public void incKount() {++w ; }
  public void P() {System.out.println("Num move = " + w);}
  TowerOfHanoi(int n, boolean recursive) {
    resetKount();
    if (recursive) {
      th_r(n,1,2,3);
    }else {
      th(n,1,2,3);
    }
  }

  private void th_r(int n, int src, int aux, int dst) {
    if (n > 0) { //at least one disk
      th_r(n-1,src,dst,aux) ; //Move n-1 from 'src' to 'aux'(using dst)
      incKount();
      System.out.println(src + "->"+ dst);
      th_r(n-1,aux,src,dst) ; //move n-1 from 'aux' to 'dst'(using src)
    }
  }
  private void th(int n, int src, int aux, int dst) {
       //WRITE YOUR CODE
  }
  private static void testBench() {
      for (int i = 1; i < 10; ++i) {
        System.out.println("-------- " + i + "---------------" );
        {
          TowerOfHanoi t = new TowerOfHanoi(i,true);
          t.P();
        }
        {
          TowerOfHanoi t = new TowerOfHanoi(i,false);
          t.P();
        }
      }
    }

  public static void main(String[] args) {
      System.out.println("TowerOfHanoi.java");
      testBench();
      System.out.println("TowerOfHanoi.java DONE");
    }
  }
```

Figure 3.22: Tower of Hanoi using stack

Problem 3.12.6. Solve the following recursion equations using **Master Theorem**

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,
where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

(1) $T(n) = 16\,T(n/4) + n$

(2) $T(n) = 4T(n/2) + n^2$

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

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

(5) $T(n) = T(n/2) + n^2$

(6) $T(n) = 2^n T(n/2) + n^n$

(7) $T(n) = 2T(n/2) + n \log n$

(8) $T(n) = 2T(n/4) + n^{0.5}$

(9) $T(n) = 0.5\,T(n/2) + 1/n$

$T(n) = aT(\frac{n}{b}) + \theta(n^k log^p n)$,

where $a \geq 1, b > 1, k \geq 0$

p is a real number

1) If $a > b^k$, then $T(n) = \theta(n^{log_b^a})$
2) If $a = b^k$
   a. If $p > -1$, then $T(n) = \theta(n^{log_b^a} log^{p+1} n)$
   b. If $p = -1$, then $T(n) = \theta(n^{log_b^a} loglogn)$
   c. If $p < -1$, then $T(n) = \theta(n^{log_b^a})$
3) If $a < b^k$
   a. If $p \geq 0$, then $T(n) = \theta(n^k log^p n)$
   b. If $p < 0$, then $T(n) = O(n^k)$

10) $T(n) = 6T(n/3) + n^2 \log n$

11) $T(n) = 64T(n/8) - n^2 \log n$

12) $T(n) = 7T(n/3) + n^2$

13) $T(n) = 4T(n/2) + \log n$

14) $T(n) = \sqrt{2}\,T(n/2) + \log n$

15) $T(n) = 2T(n/2) + \sqrt{n}$

16) $T(n) = 3T(n/2) + n$

17) $T(n) = 3T(n/3) + \sqrt{n}$

18) $T(n) = 4T(n/2) + cn$

19) $T(n) = 3T(n/4) + n \log n$