DIVIDE-AND-CONQUER

Divide-and-Conquer paradigm

In divide-and-conquer ,the problem is solved by applying the 3 steps at each level of recursion

- **Divide** the problem into a number of subproblems that are smaller instances of the same problem.
- **Conquer** the subproblems by solving them recursively. If the subproblem sizes are small enough, however, just solve the subproblems in a straightforward manner.
- **Combine** the solutions to the subproblems into the solution for the original problem.

Recurrences are used to characterize the running times of divide-and –conquer algorithms.
A ***recurrence*** is an equation or inequality that describes a function in terms of its value on smaller inputs.

- If the problem size is small enough, say n <= c for some constant c, the straightforward solution takes constant time, which we write as Θ(1).
- Suppose that our division of the problem yields a subproblems, each of which is n/b the size of the original and so it takes time aT(n/b) to solve a of them.
- If we take D(n) time to divide the problem into subproblems and C(n) time to combine the solutions to the subproblems into the solution to the original problem, we get the recurrence

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \leq c \,, \\ aT(n/b) + D(n) + C(n) & \text{otherwise} \,. \end{cases}$$

Merge sort

The ***merge sort*** algorithm follows the divide-and-conquer paradigm. Intuitively, it operates as follows.

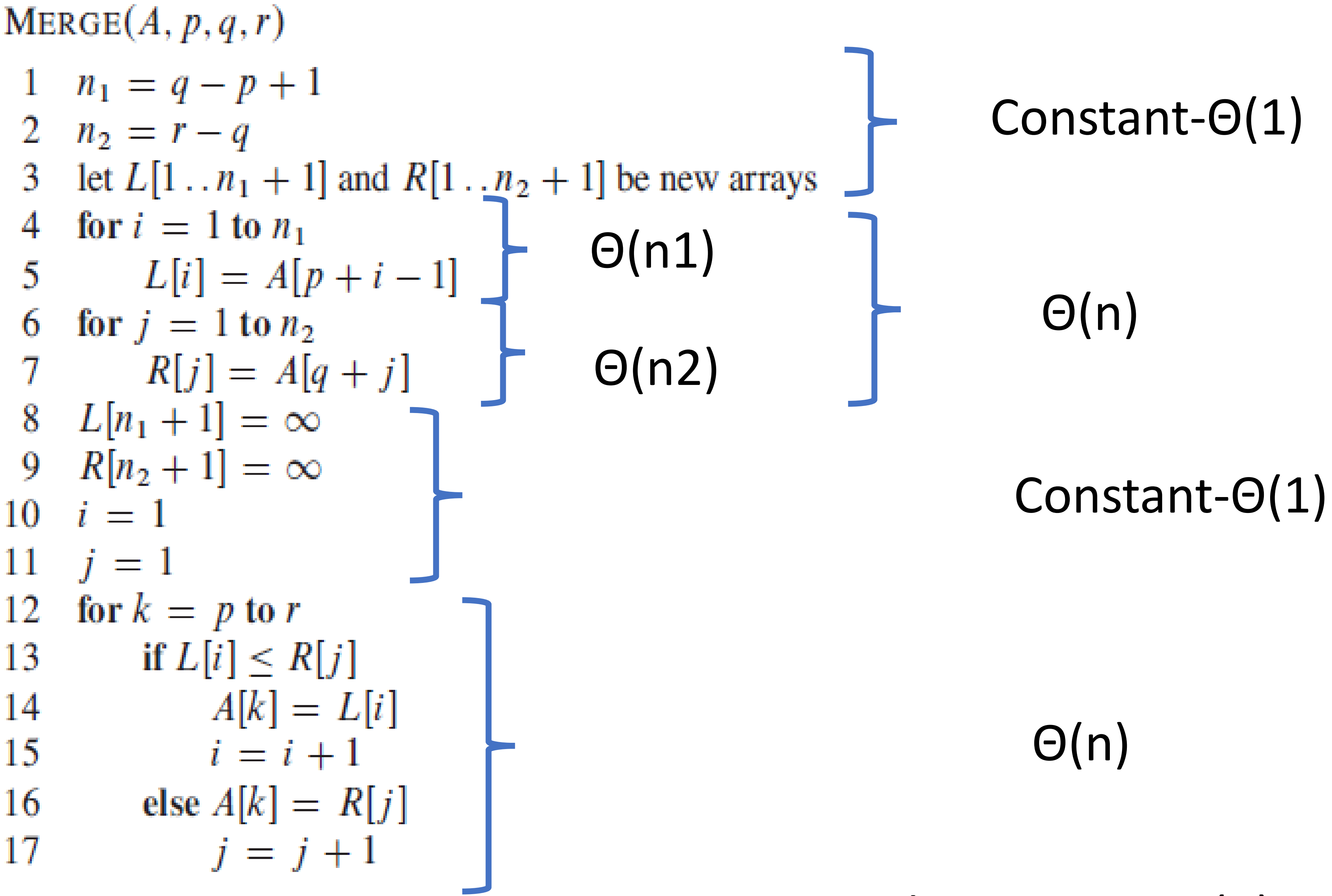
Divide: Divide the n-element sequence to be sorted into two subsequences of n/2 elements each.

Conquer: Sort the two subsequences recursively using merge sort.

Combine: Merge the two sorted subsequences to produce the sorted answer.

```
MERGE-SORT($A, p, r$)
if $p < r$
    $q = \lfloor (p + r)/2 \rfloor$
    MERGE-SORT($A, p, q$)
    MERGE-SORT($A, q + 1, r$)
    MERGE($A, p, q, r$)
```

```
MERGE($A, p, q, r$)
$n_1 = q - p + 1$
$n_2 = r - q$
let $L[1 \,..\, n_1 + 1]$ and $R[1 \,..\, n_2 + 1]$ be new arrays
for $i = 1$ to $n_1$
    $L[i] = A[p + i - 1]$
for $j = 1$ to $n_2$
    $R[j] = A[q + j]$
$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1$
$j = 1$
for $k = p$ to $r$
    if $L[i] \le R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else $A[k] = R[j]$
        $j = j + 1$
```



```
MERGE($A, p, q, r$)
$n_1 = q - p + 1$
$n_2 = r - q$
let $L[1 \mathinner{..} n_1 + 1]$ and $R[1 \mathinner{..} n_2 + 1]$ be new arrays
for $i = 1$ to $n_1$
    $L[i] = A[p + i - 1]$
for $j = 1$ to $n_2$
    $R[j] = A[q + j]$
$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1$
$j = 1$
for $k = p$ to $r$
    if $L[i] \le R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else $A[k] = R[j]$
        $j = j + 1$
```

Constant-Θ(1)

Θ(n1)

Θ(n2)

Θ(n)

Constant-Θ(1)

Θ(n)

Merge procedure runs in Θ(n)

Recurrence Relation

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \le c \text{ ,} \\ aT(n/b) + D(n) + C(n) & \text{otherwise .} \end{cases}$$

Divide: compute the middle of the subarray, which takes constant time. Thus D(n)= Θ(1).

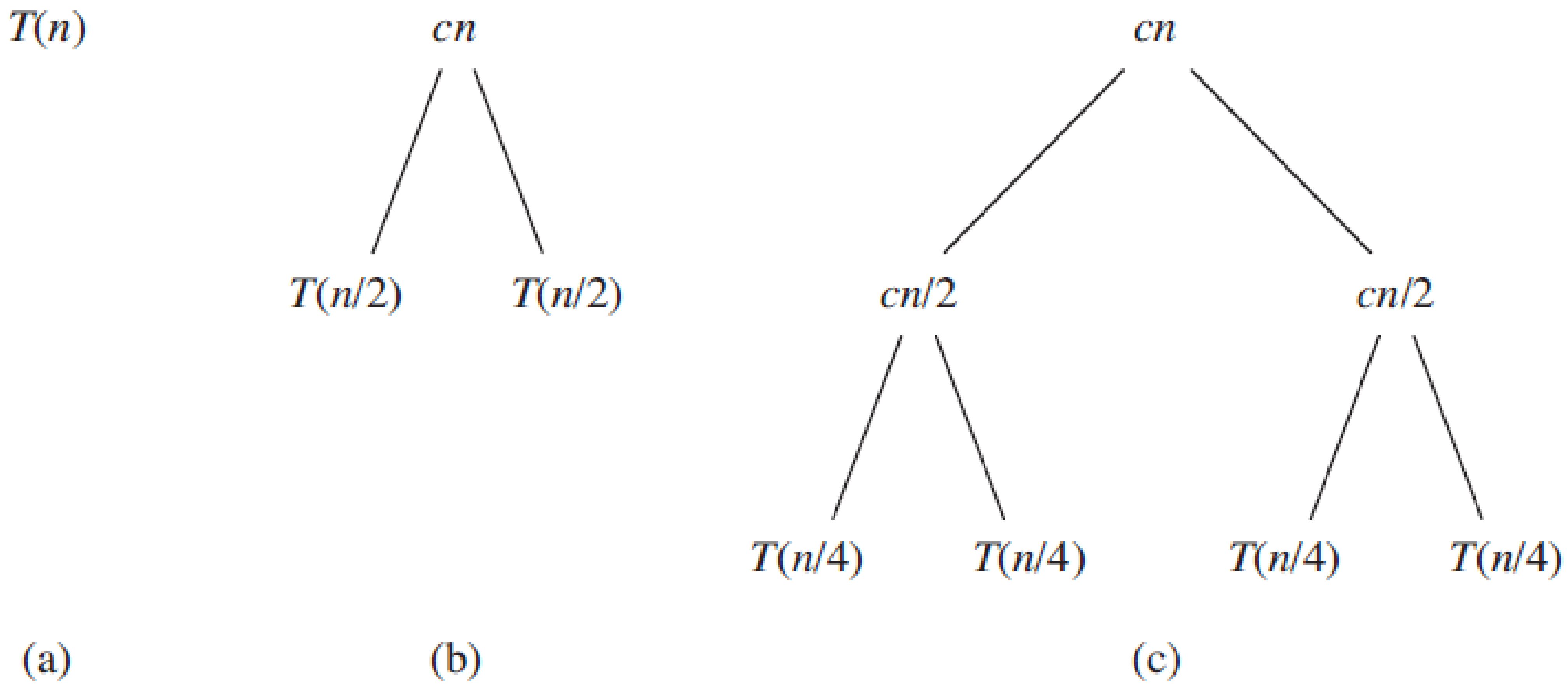
Conquer: Recursively solve the two subproblems of size n/2, which contains 2T(n/2) to the running time.

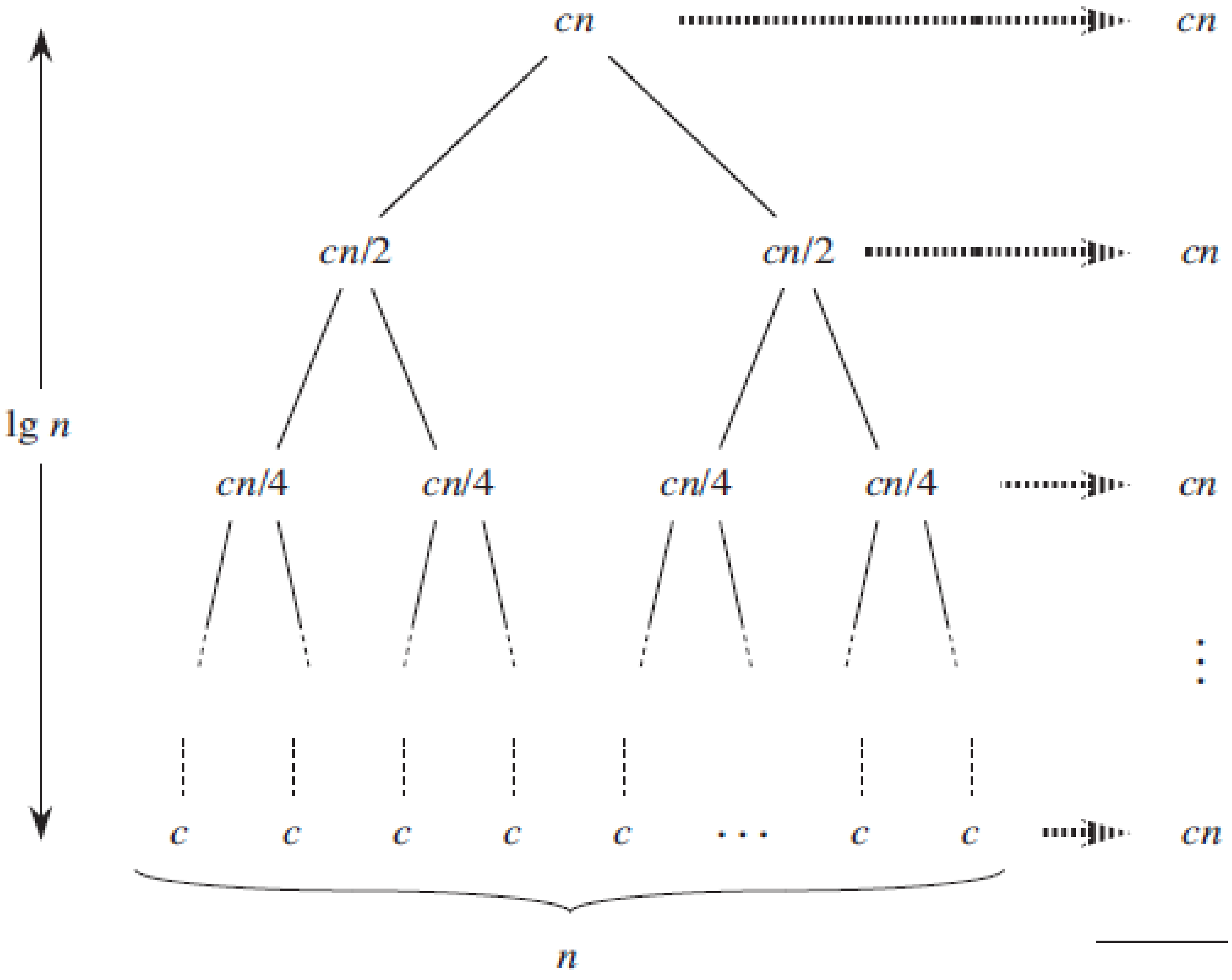
Combine: Merge procedure on an n-element subarray takes time Θ(n) and so C(n)= Θ(n)

$$T(n) = \begin{cases} c & \text{if } n = 1 \text{ ,} \\ 2T(n/2) + cn & \text{if } n > 1 \text{ ,} \end{cases}$$

Solve the recursion relation

Recursion tree




lg n
cn
$cn/2$
$cn/2$
$cn/4$
$cn/4$
$cn/4$
$cn/4$
c c c c c … c c
n
cn
cn
cn
cn
Total: $cn \lg n + cn$

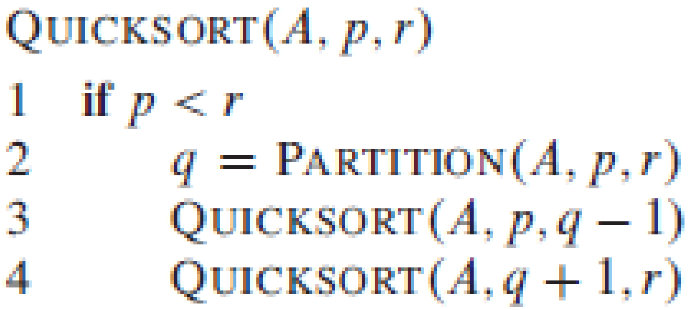
(d)

Quick sort

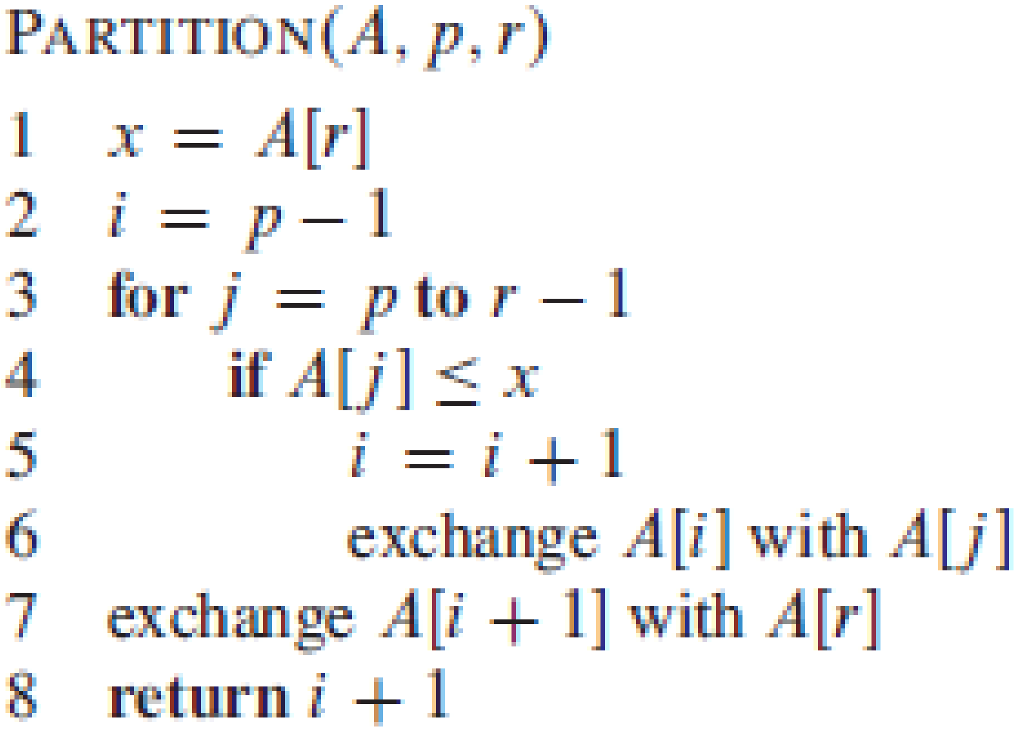
Divide: Partition (rearrange) the array A[p..r] into two (possibly empty) subarrays A[p..q-1] and A[q+1..r] such that each element of A[p..q-1] is less than or equal to A[q], which is, in turn, less than or equal to each element of A[q+1..r]. Compute the index q as part of this partitioning procedure.

Conquer: Sort the two subarrays A[p..q-1] and A[q+1..r] by recursive calls to quicksort.

Combine: Because the subarrays are already sorted, no work is needed to combine them: the entire array A[p..r] is now sorted.

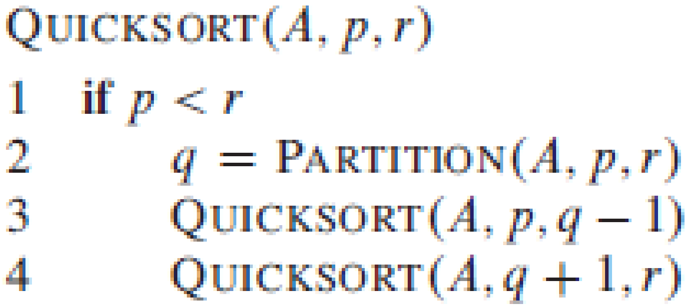


```
QUICKSORT($A, p, r$)
if $p < r$
    $q$ = PARTITION($A, p, r$)
    QUICKSORT($A, p, q - 1$)
    QUICKSORT($A, q + 1, r$)
```

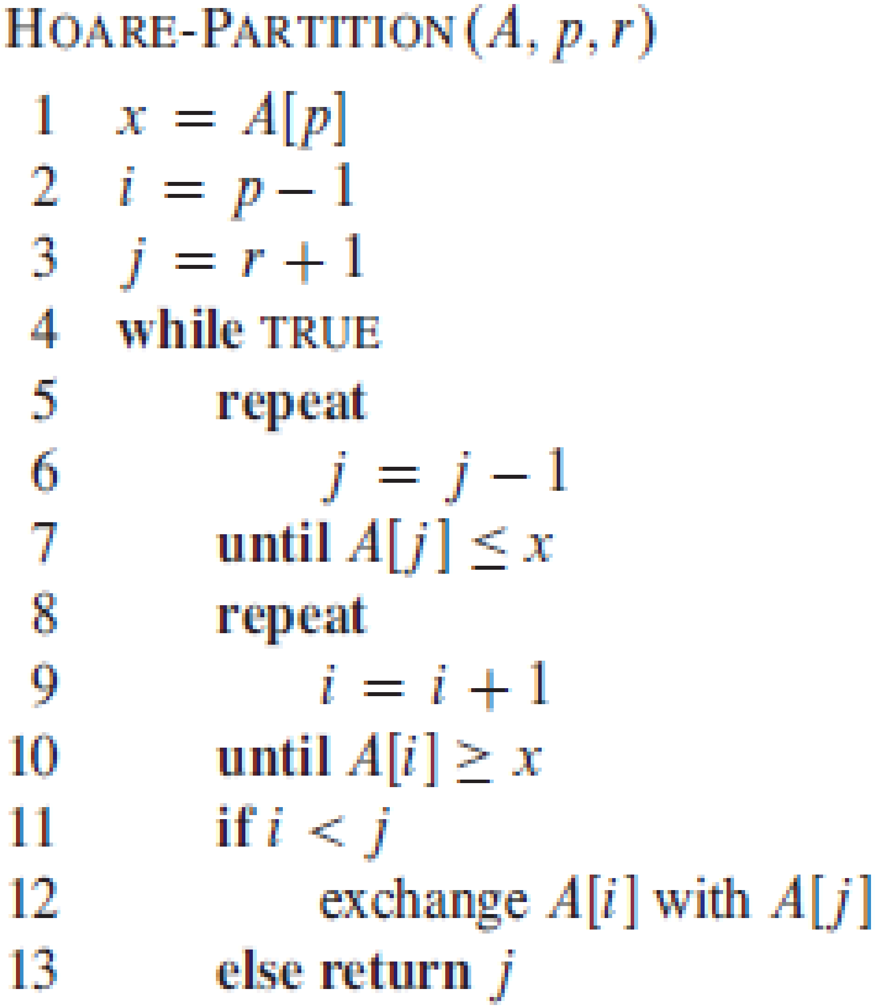


```
PARTITION($A, p, r$)
$x = A[r]$
$i = p - 1$
for $j = p$ to $r - 1$
    if $A[j] \leq x$
        $i = i + 1$
        exchange $A[i]$ with $A[j]$
exchange $A[i + 1]$ with $A[r]$
return $i + 1$
```

Quick sort Algorithm



```
QUICKSORT($A, p, r$)
if $p < r$
    $q = $ PARTITION($A, p, r$)
    QUICKSORT($A, p, q - 1$)
    QUICKSORT($A, q + 1, r$)
```



```
HOARE-PARTITION($A, p, r$)
$x = A[p]$
$i = p - 1$
$j = r + 1$
while TRUE
    repeat
        $j = j - 1$
    until $A[j] \leq x$
    repeat
        $i = i + 1$
    until $A[i] \geq x$
    if $i < j$
        exchange $A[i]$ with $A[j]$
    else return $j$
```

Performance Analysis

- The worst-case behavior for quicksort occurs when the partitioning routine produces one subproblem with n -1 elements and one with 0 elements.

Recursive call of array size n-1 is $T(n-1)$

Recursive call of array size 0 is $T(0)$ is $\Theta(1)$

Partition takes $\Theta(n)$

$$\begin{aligned} T(n) &= T(n-1) + T(0) + \Theta(n) \\ &= T(n-1) + \Theta(n) \, . \end{aligned}$$

The running time is $\Theta(n^2)$

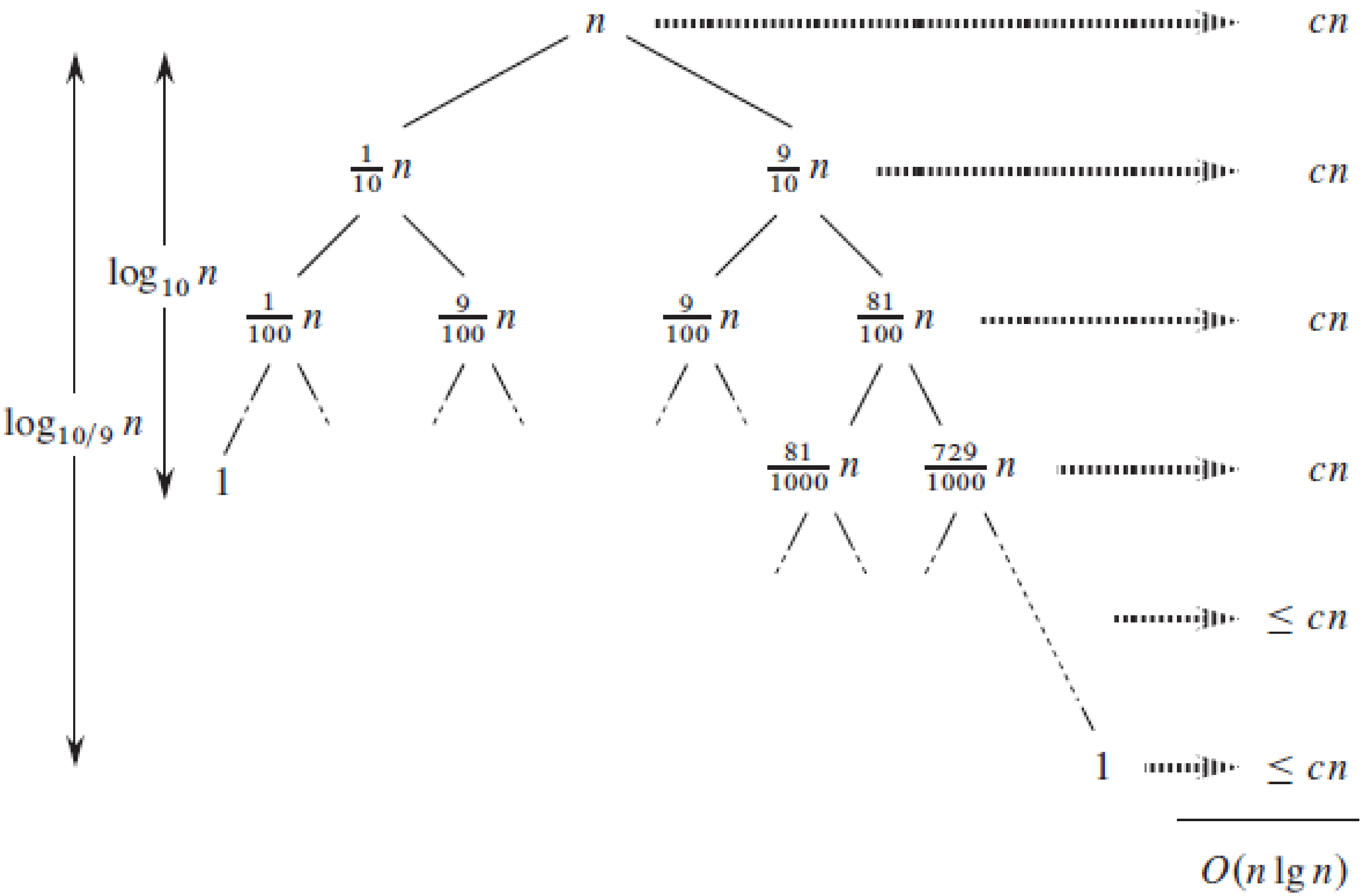
Best-case partitioning

- PARTITION produces two subproblems, each of size no more than n/2, since one is of size $\lfloor n/2 \rfloor$ and one of size $\lceil n/2 \rceil - 1$.

$$T(n) = 2T(n/2) + \Theta(n) ,$$

The running time is $\Theta(n \log n)$

Balanced partitioning



Finding the Maximum and Minimum

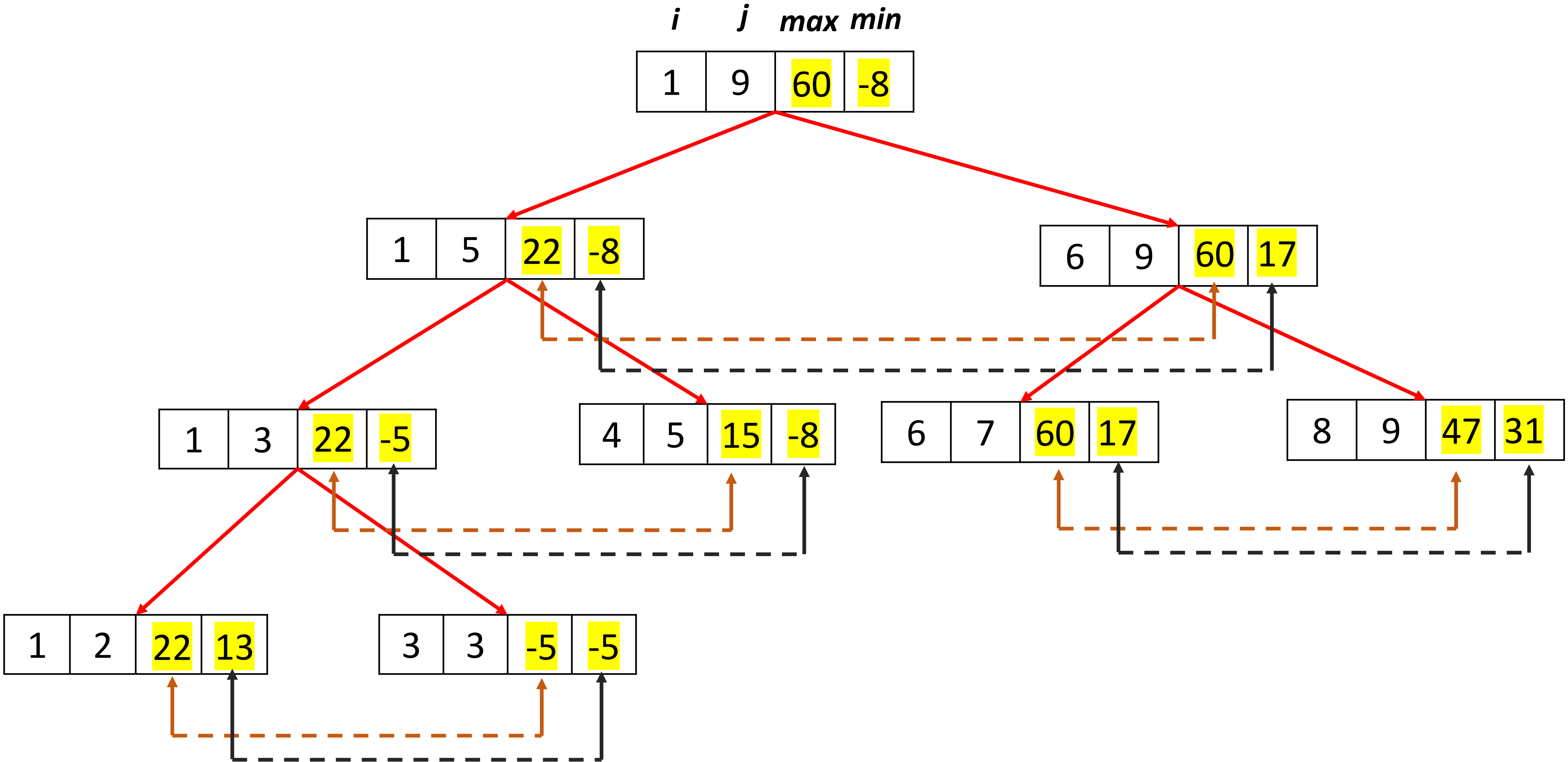
```
Algorithm StraigthMaxMin (a, n, max, min)
// set max to the maximum and min to the minimum of A[1:n]
{
        max := min:= a[1];
        for i: 2 to n do
        {
            if (a[i] >max then max:=a[i];
            if a[i] < min then min:=a[i];
        }
}
```

StraightMaxMin requires 2*(n -1) element comparisons in the best, average , and worst case.

Maximum and Minimum using Divide-and Conquer

```
Algorithm MaxMin (a, i, j, Max, Min)
    // i and j are the lower and upper bounds of an array 'a'. Max and ain contains the maximum
    // and minimum elements of an array 'a'
{     if ( i = = j) then max:=min:= a[ i ]; // Small(P)
      else if ( i == j -1 ) then // another case of Small(P)
      {
          if ( a[ i ] < a[ j ] ) then
              max: = a[ j ] ; min:=a[ i ];
          else
              max: = a[ i ] ; min:=a[ j ];
      }
      else  {
          // If P is not small divide P into subproblems .Find where to split the set
          Mid:= floor((i+j)/2)
          // solve the subproblems
          MaxMin( i ,mid,max,min);
          MaxMin(mid+1, j ,max1,min1);
          // Combine the solutions
          if (max<max1) then max:=max1;
          if (min>min1) then min:=min1;
      }
}
```

22	**13**	**-5**	**-8**	**15**	**60**	**17**	**31**	**47**
1	2	3	4	5	6	7	8	9



Matrix Multiplication

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \mathbf{X} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$$

$$c_{12} = a_{11} * b_{12} + a_{12} * b_{22}$$

$$c_{21} = a_{21} * b_{11} + a_{22} * b_{21}$$

$$c_{22} = a_{21} * b_{11} + a_{22} * b_{21}$$

Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{cc|cc} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ \hline C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{array}\right] = \left[\begin{array}{cc|cc} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ \hline A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{array}\right] \mathbf{X} \left[\begin{array}{cc|cc} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ \hline B_{31} & B_{32} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{array}\right]$$

n/2

$$\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right] = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \mathbf{X} \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

n/2

Each recursive call multiplies two n/2 X n/2 matrices.

$$C_{11} = A_{11} * B_{11} + A_{12} * B_{21}$$

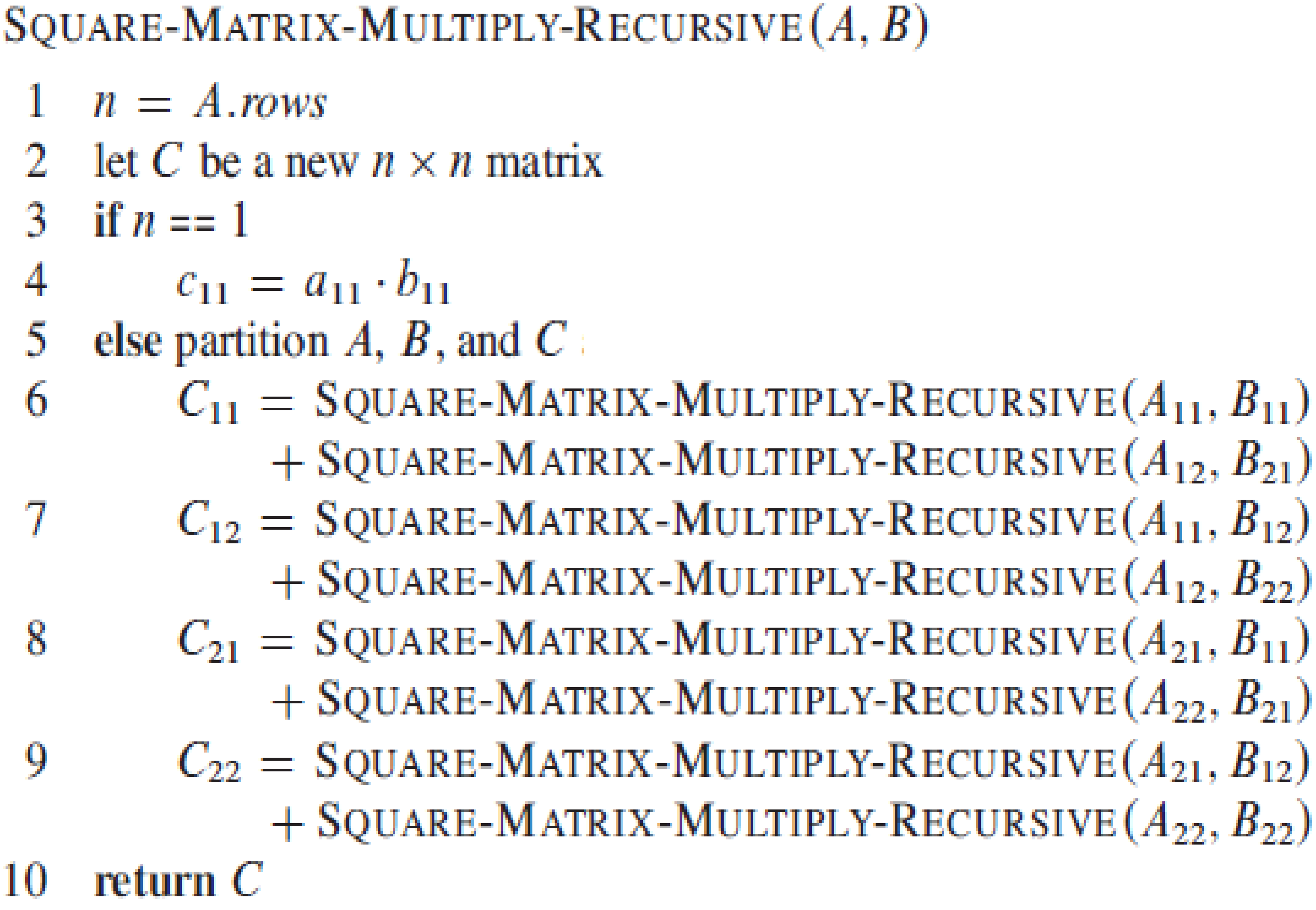
$$C_{12} = A_{11} * B_{12} + A_{12} * B_{22}$$

$$C_{21} = A_{21} * B_{11} + A_{22} * B_{21}$$

$$C_{22} = A_{21} * B_{11} + A_{22} * B_{21}$$

Matrix Multiplications : 8
Matrix Additions : 4

Matrix Multiplication Algorithm



```
SQUARE-MATRIX-MULTIPLY-RECURSIVE($A, B$)
$n = A.rows$
let $C$ be a new $n \times n$ matrix
if $n == 1$
    $c_{11} = a_{11} \cdot b_{11}$
else partition $A$, $B$, and $C$
    $C_{11} =$ SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{11}, B_{11}$)
        + SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{12}, B_{21}$)
    $C_{12} =$ SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{11}, B_{12}$)
        + SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{12}, B_{22}$)
    $C_{21} =$ SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{21}, B_{11}$)
        + SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{22}, B_{21}$)
    $C_{22} =$ SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{21}, B_{12}$)
        + SQUARE-MATRIX-MULTIPLY-RECURSIVE($A_{22}, B_{22}$)
return $C$
```

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ 8T(n/2) + \Theta(n^2) & \text{if } n > 1\,. \end{cases}$$

Recurrence relation

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ 8T(n/2) + \Theta(n^2) & \text{if } n > 1\,. \end{cases}$$

If n=1 only one scalar multiplication

If n>1

8 matrix multiplications of size n/2 x n/2 . Each recursive call takes T(n/2) running time. So, 8 calls need 8 T(n/2)

4 matrix additions of size n/2 x n/2 . Addition of two matrices of size n/2 x n/2 takes $n^2/4$. So, 4 additions take n^2

Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{cc|cc} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ \hline c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right] \mathbf{X} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

$$\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right] \quad \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \quad \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$

n/2

$$C_{11} \qquad A_{11} \ \mathbf{X} \ B_{11} \ + \ A_{12} \ \mathbf{X} \ B_{21}$$

$$\left[\begin{array}{c|c} c_{11} & c_{12} \\ \hline c_{21} & c_{22} \end{array}\right] = \begin{array}{c} \begin{matrix} A_{11} & A_{12} \end{matrix} \\ \left[\begin{array}{c|c} 1 & 2 \\ \hline 5 & 6 \end{array}\right] \\ \begin{matrix} A_{21} & A_{22} \end{matrix} \end{array} \mathbf{X} \begin{array}{c} \begin{matrix} B_{11} & B_{12} \end{matrix} \\ \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right] \\ \begin{matrix} B_{21} & B_{22} \end{matrix} \end{array} + \begin{array}{c} \begin{matrix} A_{11} & A_{12} \end{matrix} \\ \left[\begin{array}{c|c} 3 & 4 \\ \hline 7 & 8 \end{array}\right] \\ \begin{matrix} A_{21} & A_{22} \end{matrix} \end{array} \mathbf{X} \begin{array}{c} \begin{matrix} B_{11} & B_{12} \end{matrix} \\ \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right] \\ \begin{matrix} B_{21} & B_{22} \end{matrix} \end{array}$$

n/2

$$\begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 5 & 6 \end{bmatrix}$$

$C_{11} = 1 \ \mathbf{X} \ 1 + 2 \ \mathbf{X} \ 0$

$C_{12} = 1 \ \mathbf{X} \ 0 + 2 \ \mathbf{X} \ 1$

$C_{21} = 5 \ \mathbf{X} \ 1 + 6 \ \mathbf{X} \ 0$

$C_{22} = 5 \ \mathbf{X} \ 0 + 6 \ \mathbf{X} \ 1$

$C_{11} = 3 \ \mathbf{X} \ 0 + 4 \ \mathbf{X} \ 0$

$C_{12} = 3 \ \mathbf{X} \ 0 + 4 \ \mathbf{X} \ 0$

$C_{21} = 7 \ \mathbf{X} \ 0 + 8 \ \mathbf{X} \ 0$

$C_{22} = 7 \ \mathbf{X} \ 0 + 8 \ \mathbf{X} \ 0$

Matrix Multiplication- using divide and conquer

n/2

$$\left[\begin{array}{cc|cc} c_{11} & c_{12} & c_{13} & c_{14} \\ \multicolumn{2}{c|}{C_{11}} & \multicolumn{2}{c}{C_{12}} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ \hline c_{31} & c_{32} & c_{33} & c_{34} \\ \multicolumn{2}{c|}{C_{21}} & \multicolumn{2}{c}{C_{22}} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ \multicolumn{2}{c|}{A_{11}} & \multicolumn{2}{c}{A_{12}} \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \multicolumn{2}{c|}{A_{21}} & \multicolumn{2}{c}{A_{22}} \\ 13 & 14 & 15 & 16 \end{array}\right] \mathbf{X} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ \multicolumn{2}{c|}{B_{11}} & \multicolumn{2}{c}{B_{12}} \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \multicolumn{2}{c|}{B_{21}} & \multicolumn{2}{c}{B_{22}} \\ 0 & 0 & 0 & 1 \end{array}\right]$$

$$C_{12} \qquad A_{11} \ \mathbf{X} \ B_{12} \ + \ A_{12} \ \mathbf{X} \ B_{22}$$

n/2

$$\left[\begin{array}{c|c} c_{11} & c_{12} \\ \hline c_{21} & c_{22} \end{array}\right] = \begin{array}{cc} A_{11} & A_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 1 & 2 \\ \hline 5 & 6 \end{array}\right]} \\ A_{21} & A_{22} \end{array} \mathbf{X} \begin{array}{cc} B_{11} & B_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right]} \\ B_{21} & B_{22} \end{array} + \begin{array}{cc} A_{11} & A_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 3 & 4 \\ \hline 7 & 8 \end{array}\right]} \\ A_{21} & A_{22} \end{array} \mathbf{X} \begin{array}{cc} B_{11} & B_{12} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right]} \\ B_{21} & B_{22} \end{array}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix}$$

C_{11} = 1 **X** 0 + 2 **X** 0

C_{12} = 1 **X** 0 + 2 **X** 0

C_{21} = 5 **X** 0 + 6 **X** 0

C_{22} = 5 **X** 0 + 6 **X** 0

C_{11} = 3 **X** 1 + 4 **X** 0

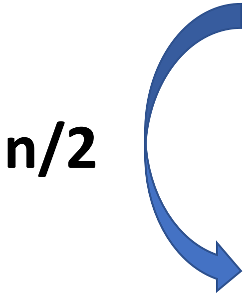
C_{12} = 3 **X** 0 + 4 **X** 1

C_{21} = 7 **X** 1 + 8 **X** 0

C_{22} = 7 **X** 0 + 8 **X** 1

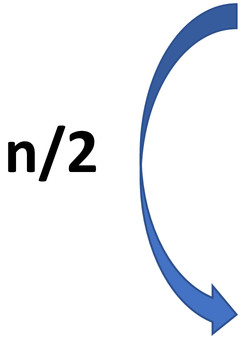
Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{c|c} \underset{C_{11}}{\begin{matrix}c_{11} & c_{12}\\ c_{21} & c_{22}\end{matrix}} & \underset{C_{12}}{\begin{matrix}c_{13} & c_{14}\\ c_{23} & c_{24}\end{matrix}} \\ \hline \underset{C_{21}}{\begin{matrix}c_{31} & c_{32}\\ c_{41} & c_{42}\end{matrix}} & \underset{C_{22}}{\begin{matrix}c_{33} & c_{34}\\ c_{43} & c_{44}\end{matrix}} \end{array}\right] = \left[\begin{array}{c|c} \underset{A_{11}}{\begin{matrix}1 & 2\\ 5 & 6\end{matrix}} & \underset{A_{12}}{\begin{matrix}3 & 4\\ 7 & 8\end{matrix}} \\ \hline \underset{A_{21}}{\begin{matrix}9 & 10\\ 13 & 14\end{matrix}} & \underset{A_{22}}{\begin{matrix}11 & 12\\ 15 & 16\end{matrix}} \end{array}\right] \mathbf{X} \left[\begin{array}{c|c} \underset{B_{11}}{\begin{matrix}1 & 0\\ 0 & 1\end{matrix}} & \underset{B_{12}}{\begin{matrix}0 & 0\\ 0 & 0\end{matrix}} \\ \hline \underset{B_{21}}{\begin{matrix}0 & 0\\ 0 & 0\end{matrix}} & \underset{B_{22}}{\begin{matrix}1 & 0\\ 0 & 1\end{matrix}} \end{array}\right]$$



$$C_{21} \qquad A_{21} \ \mathbf{X} \ B_{11} \ + \ A_{22} \ \mathbf{X} \ B_{21}$$

$$\left[\begin{array}{c|c} c_{11} & c_{12}\\ \hline c_{21} & c_{22}\end{array}\right] = \begin{array}{c}\begin{matrix}A_{11} & A_{12}\end{matrix}\\ \left[\begin{array}{c|c} 9 & 10\\ \hline 13 & 14\end{array}\right]\\ \begin{matrix}A_{21} & A_{22}\end{matrix}\end{array} \mathbf{X} \begin{array}{c}\begin{matrix}B_{11} & B_{12}\end{matrix}\\ \left[\begin{array}{c|c} 1 & 0\\ \hline 0 & 1\end{array}\right]\\ \begin{matrix}B_{21} & B_{22}\end{matrix}\end{array} + \begin{array}{c}\begin{matrix}A_{11} & A_{12}\end{matrix}\\ \left[\begin{array}{c|c} 11 & 12\\ \hline 15 & 16\end{array}\right]\\ \begin{matrix}A_{21} & A_{22}\end{matrix}\end{array} \mathbf{X} \begin{array}{c}\begin{matrix}B_{11} & B_{12}\end{matrix}\\ \left[\begin{array}{c|c} 0 & 0\\ \hline 0 & 0\end{array}\right]\\ \begin{matrix}B_{21} & B_{22}\end{matrix}\end{array}$$



$$\begin{bmatrix} 9 & 10\\ 13 & 14\end{bmatrix} + \begin{bmatrix} 0 & 0\\ 0 & 0\end{bmatrix} = \begin{bmatrix} 9 & 10\\ 13 & 14\end{bmatrix}$$

C_{11} = 9 **X** 1 + 10 **X** 0

C_{12} = 9 **X** 0 + 10 **X** 1

C_{21} = 13 **X** 1 + 14 **X** 0

C_{22} = 13 **X** 0 + 14 **X** 1

C_{11} = 11 **X** 0 + 12 **X** 0

C_{12} = 11 **X** 0 + 12 **X** 0

C_{21} = 15 **X** 0 + 16 **X** 0

C_{22} = 15 **X** 0 + 16 **X** 1

Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{cc|cc} c_{11} & c_{12} & c_{13} & c_{14} \\ \multicolumn{2}{c|}{C_{11}} & \multicolumn{2}{c}{C_{12}} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ \hline c_{31} & c_{32} & c_{33} & c_{34} \\ \multicolumn{2}{c|}{C_{21}} & \multicolumn{2}{c}{C_{22}} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ \multicolumn{2}{c|}{A_{11}} & \multicolumn{2}{c}{A_{12}} \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ \multicolumn{2}{c|}{A_{21}} & \multicolumn{2}{c}{A_{22}} \\ 13 & 14 & 15 & 16 \end{array}\right] \mathbf{X} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ \multicolumn{2}{c|}{B_{11}} & \multicolumn{2}{c}{B_{12}} \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \multicolumn{2}{c|}{B_{21}} & \multicolumn{2}{c}{B_{22}} \\ 0 & 0 & 0 & 1 \end{array}\right]$$

n/2

$$C_{22} \qquad A_{21} \; \mathbf{X} \; B_{12} \; + \; A_{22} \; \mathbf{X} \; B_{22}$$

$$\left[\begin{array}{c|c} c_{11} & c_{12} \\ \hline c_{21} & c_{22} \end{array}\right] = \begin{array}{cc} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 9 & 10 \\ \hline 13 & 14 \end{array}\right]} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{array} \mathbf{X} \begin{array}{cc} \mathbf{B_{11}} & \mathbf{B_{12}} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right]} \\ \mathbf{B_{21}} & \mathbf{B_{22}} \end{array} + \begin{array}{cc} \mathbf{A_{11}} & \mathbf{A_{12}} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 11 & 12 \\ \hline 15 & 16 \end{array}\right]} \\ \mathbf{A_{21}} & \mathbf{A_{22}} \end{array} \mathbf{X} \begin{array}{cc} \mathbf{B_{11}} & \mathbf{B_{12}} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right]} \\ \mathbf{B_{21}} & \mathbf{B_{22}} \end{array}$$

n/2

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 11 & 12 \\ 15 & 16 \end{bmatrix} = \begin{bmatrix} 11 & 12 \\ 15 & 16 \end{bmatrix}$$

$C_{11} = 9 \;\mathbf{X}\; 0 + 10 \;\mathbf{X}\; 0$

$C_{12} = 9 \;\mathbf{X}\; 0 + 10 \;\mathbf{X}\; 0$

$C_{21} = 13 \;\mathbf{X}\; 0 + 14 \;\mathbf{X}\; 0$

$C_{22} = 13 \;\mathbf{X}\; 0 + 14 \;\mathbf{X}\; 0$

$C_{11} = 11 \;\mathbf{X}\; 1 + 12 \;\mathbf{X}\; 0$

$C_{12} = 11 \;\mathbf{X}\; 0 + 12 \;\mathbf{X}\; 1$

$C_{21} = 15 \;\mathbf{X}\; 1 + 16 \;\mathbf{X}\; 0$

$C_{22} = 15 \;\mathbf{X}\; 0 + 16 \;\mathbf{X}\; 1$

Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{cc|cc} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ \hline C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right] \mathbf{X} \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

n/2

$$\left[\begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array}\right] = \left[\begin{array}{c|c} 1 & 2 \\ \hline 5 & 6 \end{array}\right] \mathbf{X} \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{cc} 1 & 2 \\ 5 & 6 \end{array}\right]$$

n/2

$C_{11} = 1 \; \mathbf{X} \; 1 + 2 \; \mathbf{X} \; 0$

$C_{12} = 1 \; \mathbf{X} \; 0 + 2 \; \mathbf{X} \; 1$

$C_{21} = 5 \; \mathbf{X} \; 1 + 6 \; \mathbf{X} \; 0$

$C_{22} = 5 \; \mathbf{X} \; 0 + 6 \; \mathbf{X} \; 1$

Performance Analysis

$T(n) = 8 * T(n/2) + cn^2$

$= 8 * [\ 8* T(n/4) + cn^2/4\ ] + cn^2$

$= 2^6 * T(n/4) + 2cn^2 + cn^2$

$= 2^6 * [8 * T(n/8) + cn^2/8\ ] + 2cn^2 + cn^2$

$= 2^9 * T(n/8) + 4cn^2 + 2cn^2 + cn^2$

…

…

$= (2^3)^k * T(n/2^K) + (2^K - 1)\ cn^2$

$= (8)^k * T(n/2^K) + (2^K - 1)\ cn^2$

Let $2^K = n$ then $k = \log_2 n$

$= (8)^{\log_2 n} * T(n/n) + (n - 1)\ cn^2$

$= (2^3)^{\log_2 n} * T(n/n) + (n - 1)\ cn^2$

$= (2)^{\log_2 n^3} + (n - 1)\ cn^2$

$= n^3 + (n - 1)\ cn^2$

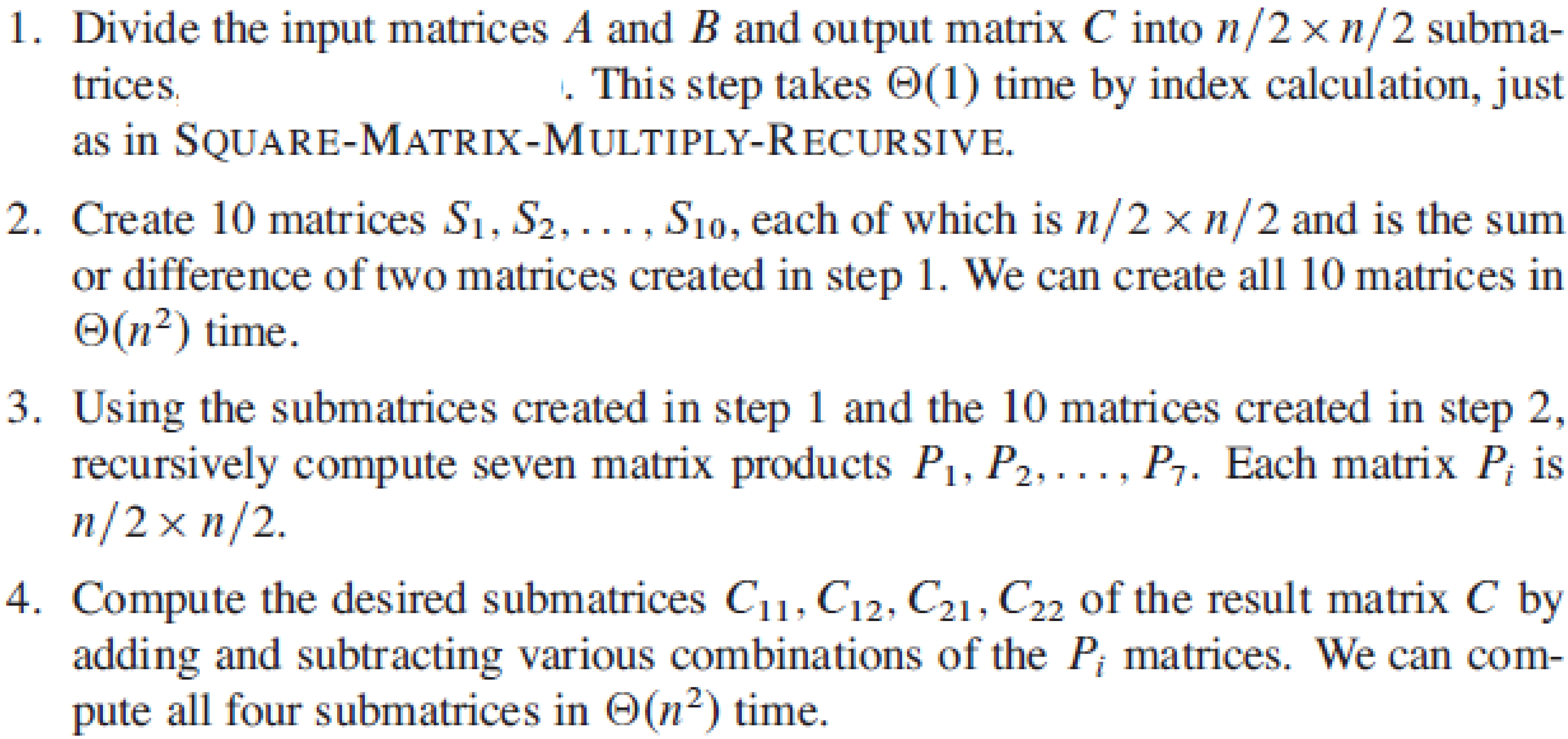
$= \Theta\ (n^3)$

Strassen’s Matrix Multiplication

- The key to Strassen’s method is to perform seven recursive multiplications instead of performing eight recursive multiplications of n/2 x n/2.
- The cost of eliminating one matrix multiplication will create several new additions of n/2 x n/2 matrices, but still only a constant number .

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1\,, \\ 7T(n/2) + \Theta(n^2) & \text{if } n > 1\,. \end{cases}$$

Strassen’s matrix multiplication method



1. Divide the input matrices A and B and output matrix C into $n/2 \times n/2$ submatrices, . This step takes $\Theta(1)$ time by index calculation, just as in SQUARE-MATRIX-MULTIPLY-RECURSIVE.
2. Create 10 matrices $S_1, S_2, \ldots, S_{10}$, each of which is $n/2 \times n/2$ and is the sum or difference of two matrices created in step 1. We can create all 10 matrices in $\Theta(n^2)$ time.
3. Using the submatrices created in step 1 and the 10 matrices created in step 2, recursively compute seven matrix products $P_1, P_2, \ldots, P_7$. Each matrix P_i is $n/2 \times n/2$.
4. Compute the desired submatrices $C_{11}, C_{12}, C_{21}, C_{22}$ of the result matrix C by adding and subtracting various combinations of the P_i matrices. We can compute all four submatrices in $\Theta(n^2)$ time.

Step 2

$$
\begin{aligned}
S_1 &= B_{12} - B_{22} ,\\
S_2 &= A_{11} + A_{12} ,\\
S_3 &= A_{21} + A_{22} ,\\
S_4 &= B_{21} - B_{11} ,\\
S_5 &= A_{11} + A_{22} ,\\
S_6 &= B_{11} + B_{22} ,\\
S_7 &= A_{12} - A_{22} ,\\
S_8 &= B_{21} + B_{22} ,\\
S_9 &= A_{11} - A_{21} ,\\
S_{10} &= B_{11} + B_{12} .
\end{aligned}
$$

Step 3

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22} ,\\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22} ,\\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11} ,\\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11} ,\\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22} ,\\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22} ,\\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12} .
\end{aligned}
$$

Step 4

$$C_{11} = P_5 + P_4 - P_2 + P_6 .$$

$$C_{12} = P_1 + P_2$$

$$C_{21} = P_3 + P_4$$

$$C_{22} = P_5 + P_1 - P_3 - P_7 ,$$

Matrix Multiplication- using divide and conquer

$$\left[\begin{array}{cc|cc} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ \hline c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}\right] X \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

C_{11} C_{12} C_{21} C_{22} | A_{11} A_{12} A_{21} A_{22} | B_{11} B_{12} B_{21} B_{22}

n/2

$$C_{12} \qquad A_{11} \; X \; B_{12} \; + \; A_{12} \; X \; B_{22}$$

n/2

$$\left[\begin{array}{c|c} c_{11} & c_{12} \\ \hline c_{21} & c_{22} \end{array}\right] = \overset{A_{11}\;\;A_{12}}{\underset{A_{21}\;\;A_{22}}{\left[\begin{array}{c|c} 1 & 2 \\ \hline 5 & 6 \end{array}\right]}} X \overset{B_{11}\;\;B_{12}}{\underset{B_{21}\;\;B_{22}}{\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & 0 \end{array}\right]}} + \overset{A_{11}\;\;A_{12}}{\underset{A_{21}\;\;A_{22}}{\left[\begin{array}{c|c} 3 & 4 \\ \hline 7 & 8 \end{array}\right]}} X \overset{B_{11}\;\;B_{12}}{\underset{B_{21}\;\;B_{22}}{\left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & 1 \end{array}\right]}}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 7 & 8 \end{bmatrix}$$

In place of recursive calls straseen's use his formulas S_1 to S_{10}, P_1 to P_{10}, C_{11} to C_{22}

C_{11} = 1 X 0 + 2 X 0

C_{12} = 1 X 0 + 2 X 0

C_{21} = 5 X 0 + 6 X 0

C_{22} = 5 X 0 + 6 X 0

C_{11} = 3 X 1 + 4 X 0

C_{12} = 3 X 0 + 4 X 1

C_{21} = 7 X 1 + 8 X 0

C_{22} = 7 X 0 + 8 X 1

Performance Analysis of straseen’s Matrix multiplication

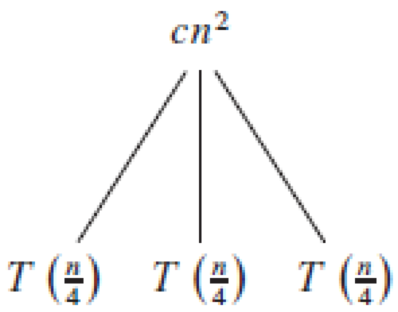
$$T(n) = 7 * T(n/2) + cn^2$$

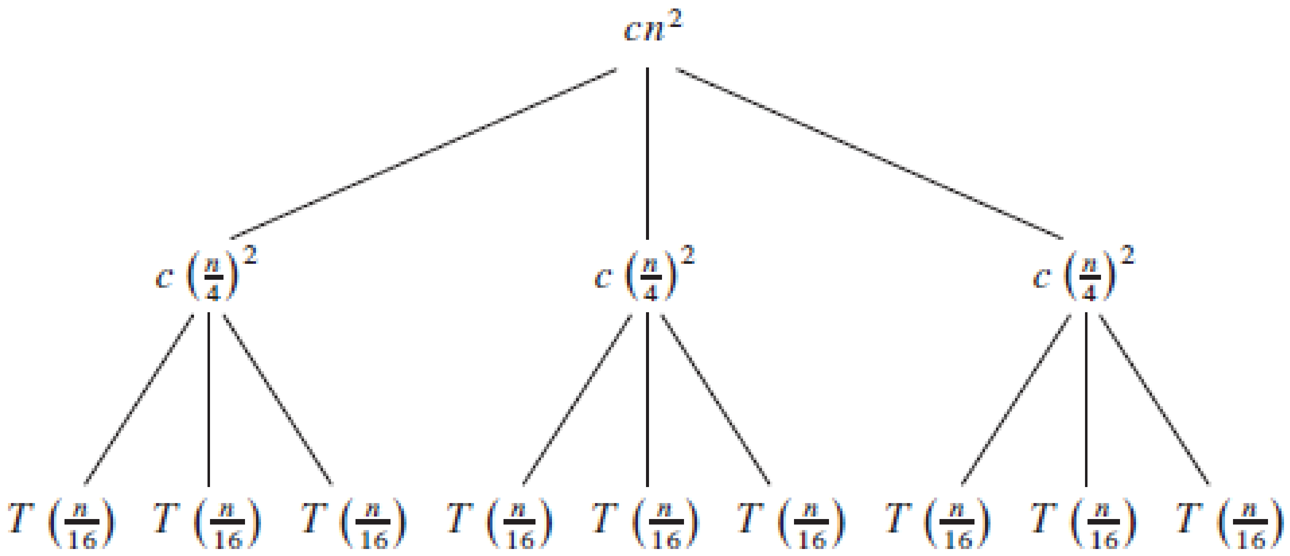
$$= 7 * [7* T(n/4) +cn^2/4]+ cn^2$$

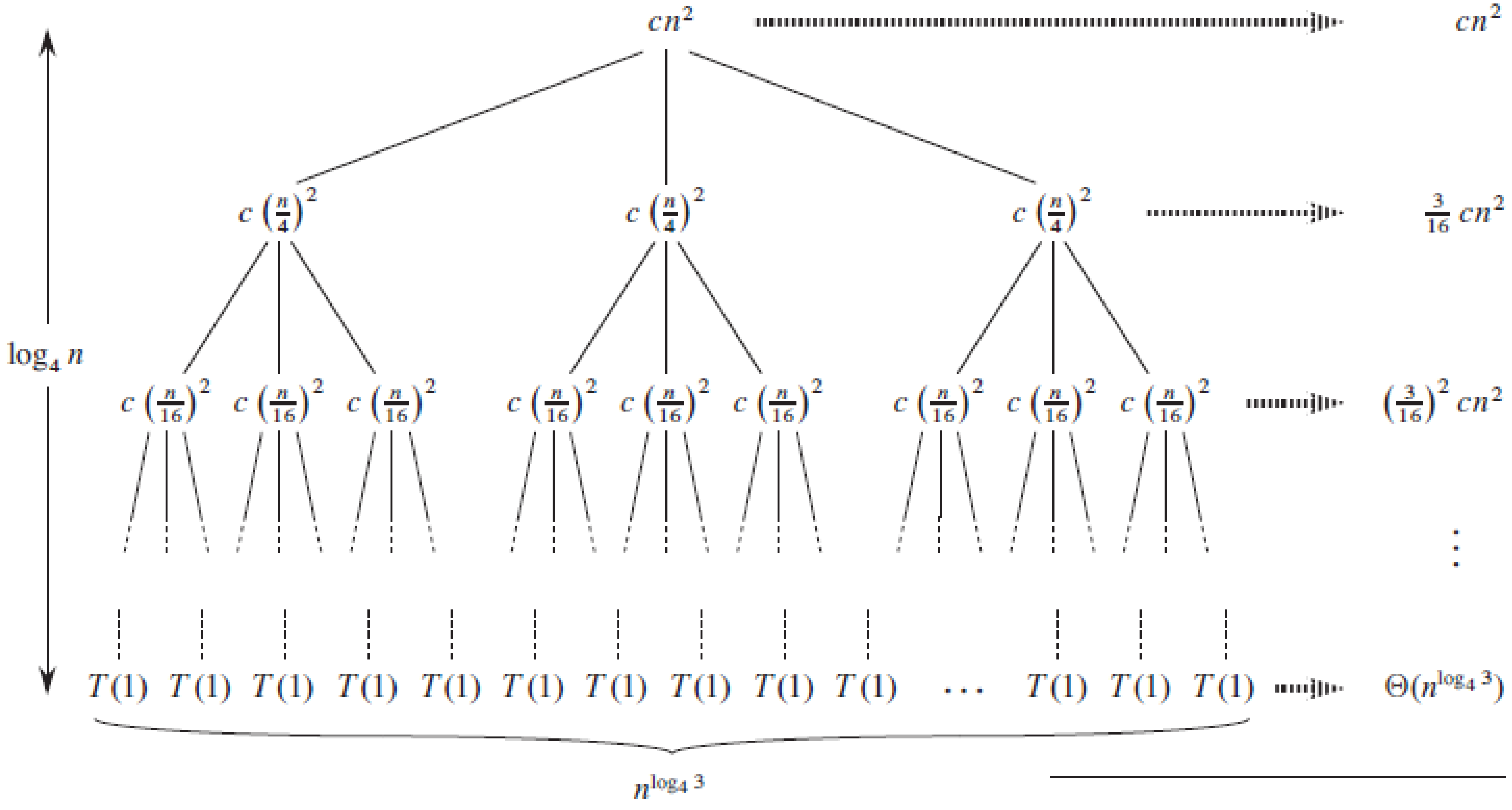
$$= 7^2 *[7 * T(n/8) + cn^2/16]+ 7cn^2 /4 + cn^2$$

Solving the Recurrence Relation using recursion tree

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

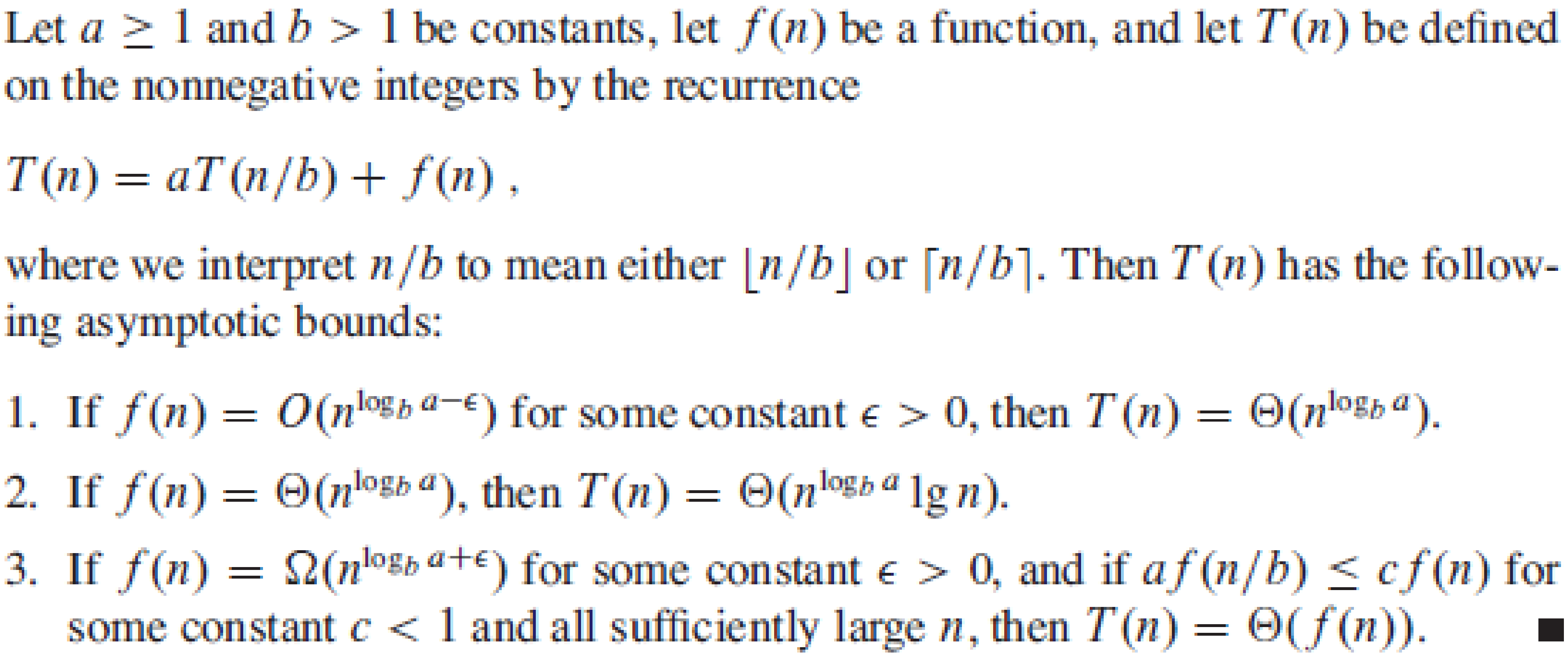

cn^2
$T\left(\frac{n}{4}\right)$
$T\left(\frac{n}{4}\right)$
$T\left(\frac{n}{4}\right)$


cn^2
$c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{4}\right)^2$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$
$T\left(\frac{n}{16}\right)$


cn^2
$c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{16}\right)^2$
$\log_4 n$
$T(1)$
$\cdots$
$n^{\log_4 3}$
cn^2
$\frac{3}{16}\,cn^2$
$\left(\frac{3}{16}\right)^2 cn^2$
$\vdots$
$\Theta(n^{\log_4 3})$
Total: $O(n^2)$

(d)

Master’s Theorem



Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n) ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. ■