- give brief justification for algorithms in homework

```
QuickSort(A[1..n]):
  if (n > 1)
      Choose a pivot element A[p]
      r ← Partition(A, p)
      QuickSort(A[1..r−1])      ⟨⟨Recurse!⟩⟩
      QuickSort(A[r+1..n])      ⟨⟨Recurse!⟩⟩
```

new index of pivot

```
Partition(A[1..n], p):
  swap A[p] ↔ A[n]
  ℓ ← 0                  ⟨⟨#items < pivot⟩⟩
  for i ← 1 to n−1
      if A[i] < A[n]
          ℓ ← ℓ+1
          swap A[ℓ] ↔ A[i]
  swap A[n] ↔ A[ℓ+1]
  return ℓ+1
```

Input:	S	O	R	T	I	N	G	E	X	A	M	P	L
Choose a pivot:	S	O	R	T	I	N	G	E	X	A	M	$\boxed{P}$	L
Partition:	A	G	O	E	I	N	L	M	$\boxed{P}$	T	X	S	R
Recurse Left:	A	E	G	I	L	M	N	O	$\boxed{P}$	T	X	S	R
Recurse Right:	A	E	G	I	L	M	N	O	$\boxed{P}$	R	S	T	X

$$T(n) = \Theta(n) + \max_{1 \le r \le n} \left(T(r-1) + T(n-r) \right)$$

$$T(n) \ge \Omega(n) + T(0) + T(n-1)$$

(if $r = 1$ or $r = n$)

$$\Rightarrow T(n) \ge \Omega(n^2)$$

$$(+\ T(n) = O(n^2))$$

magic: $r = n/2$

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

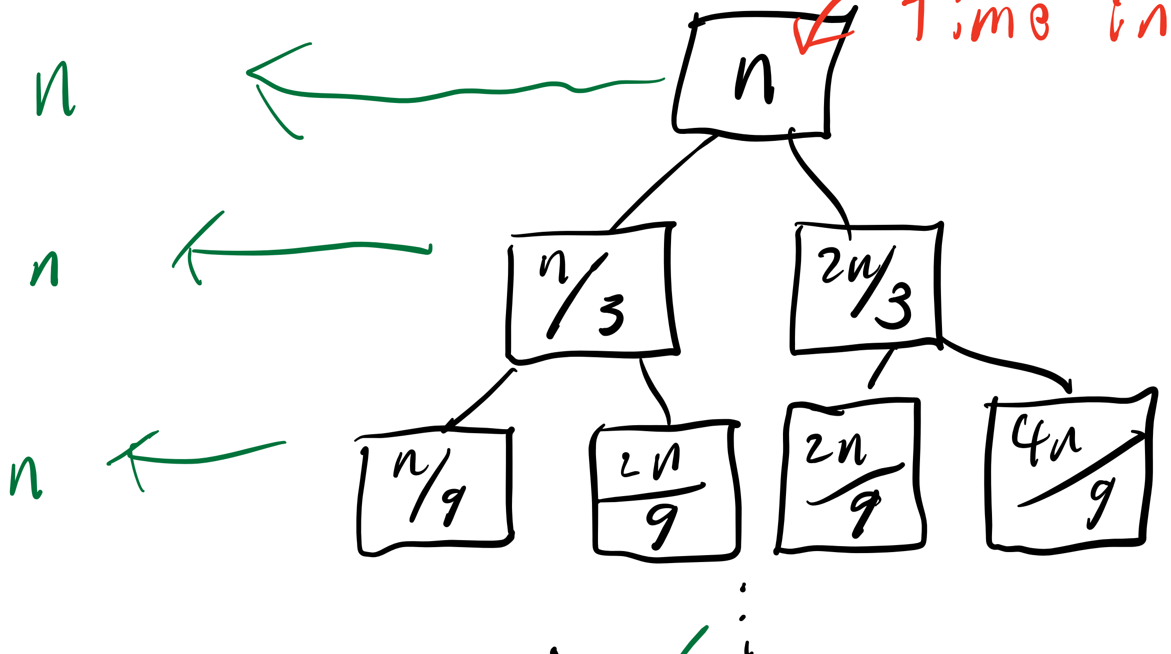
$$= O(n \log n)$$

less magic: $\frac{n}{3} \le r \le \frac{2n}{3}$

$$\Rightarrow T(n) = O(n) + \max_{n/3 \le r \le \frac{2n}{3}} \left(T(n-r) + T(r-1) \right)$$

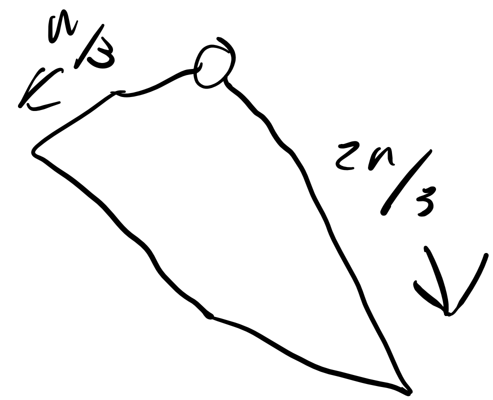
$$T(n) \ge \Omega(n) + T(n/3) + T(2n/3)$$

← time in root subproblem

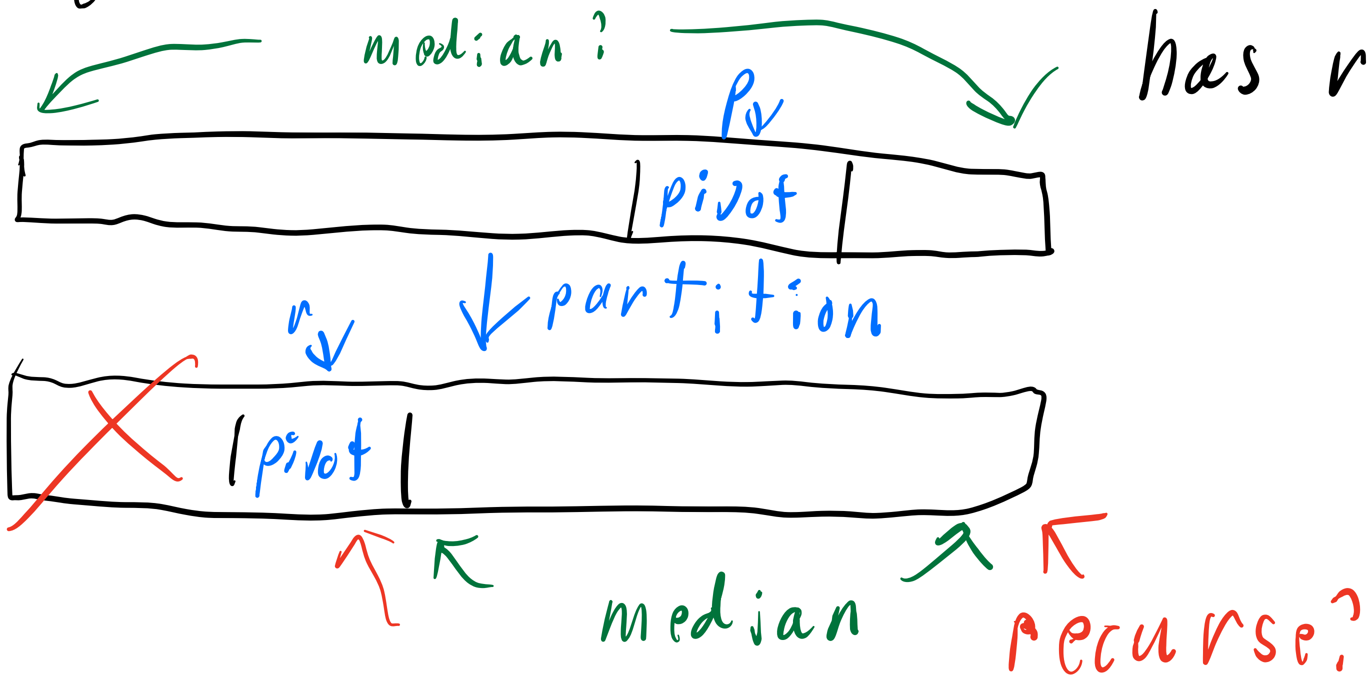


Each level $\le n$, but depth is $\log_{3/2} n$.

$T(n) = O(n \log n)$. ✓



Selection: Given $A[1..n]$, find the median,
has rank $\lfloor n/2 \rfloor$.



Given A and integer k s.t. $1 \le k \le n$:
find element of rank k (median has $k = \lfloor n/2 \rfloor$)

```
QuickSelect(A[1..n], k):
  if n = 1
      return A[1]
  else
      Choose a pivot element A[p]
      r ← Partition(A[1..n], p)

      if k < r
```

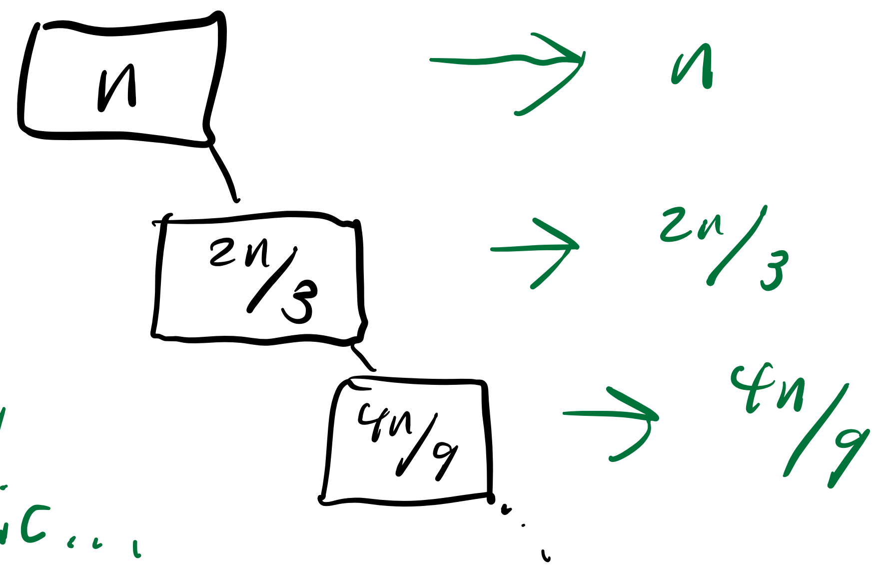
```
        return QuickSelect(A[1 .. r − 1], k)
    else if k > r
        return QuickSelect(A[r + 1 .. n], k − r)
    else
        return A[r]
```

$$T(n) \leq O(n) + \max_{1 \leq r \leq n} \left(\max\{T(r-1), T(n-r)\} \right)$$

If $r=1$.. $T(n) \leq O(n) + T(n-1)$

$$= O(n^2)$$

less magic $n/3 \leq n \leq 2n/3$ $T(n) = O(n) + T(2n/3)$



decreasing geometric...

proportional to largest term

$T(n) = O(n)$

Blum, Floyd, Pratt, Rivest, + Tarjan early 70's:

choose pivot using recursion

(want a middle element from a "representive" sample)

✓ "median of medians" selection

MomSelect($A[1..n]$, k):
  if $n \le 25$ ⟨⟨or whatever⟩⟩
    use brute force
  else
    $m \leftarrow \lceil n/5 \rceil$
    for $i \leftarrow 1$ to m
      $M[i] \leftarrow$ MedianOfFive($A[5i-4..5i]$) ⟨⟨Brute force!⟩⟩
    $mom \leftarrow$ MomSelect($M[1..m]$, $\lfloor m/2 \rfloor$) ⟨⟨Recursion!⟩⟩
    $r \leftarrow$ Partition($A[1..n]$, ~~mom~~) index of mom
    if $k < r$

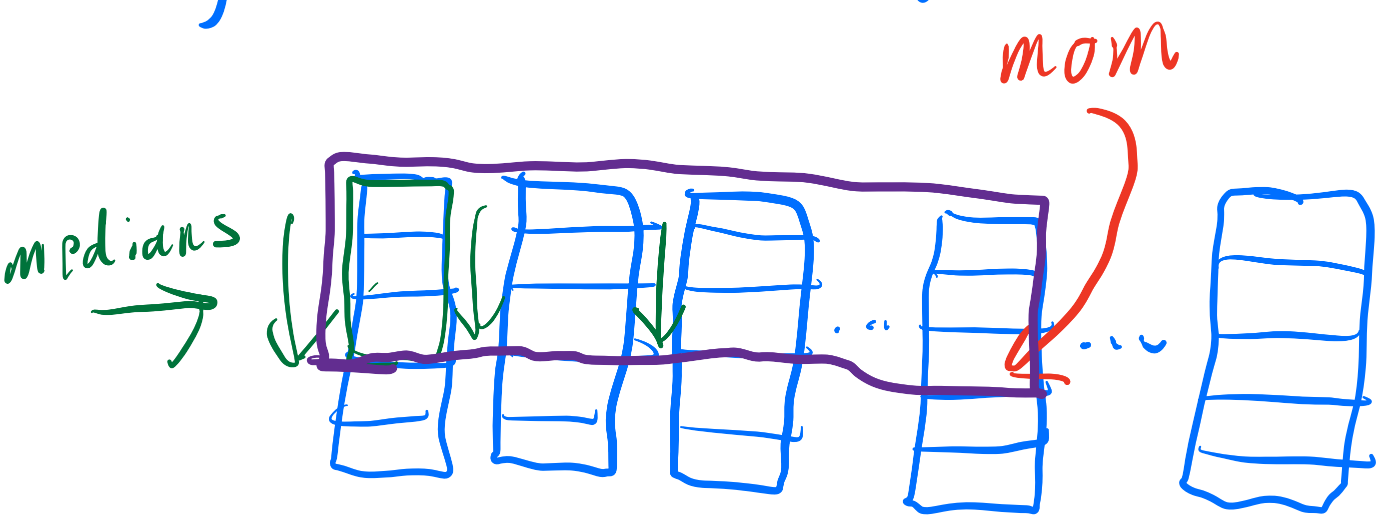
O(n)

$\lceil n/5 \rceil$ blocks of 5 elements each

```
        return MomSelect(A[1 .. r − 1], k)       ⟨⟨Recursion!⟩⟩
    else if k > r
        return MomSelect(A[r + 1 .. n], k − r)   ⟨⟨Recursion!⟩⟩
    else
        return mom
```

find median of each block

Imagine... rearrange A into a $5 \times \lceil n/5 \rceil$ grid



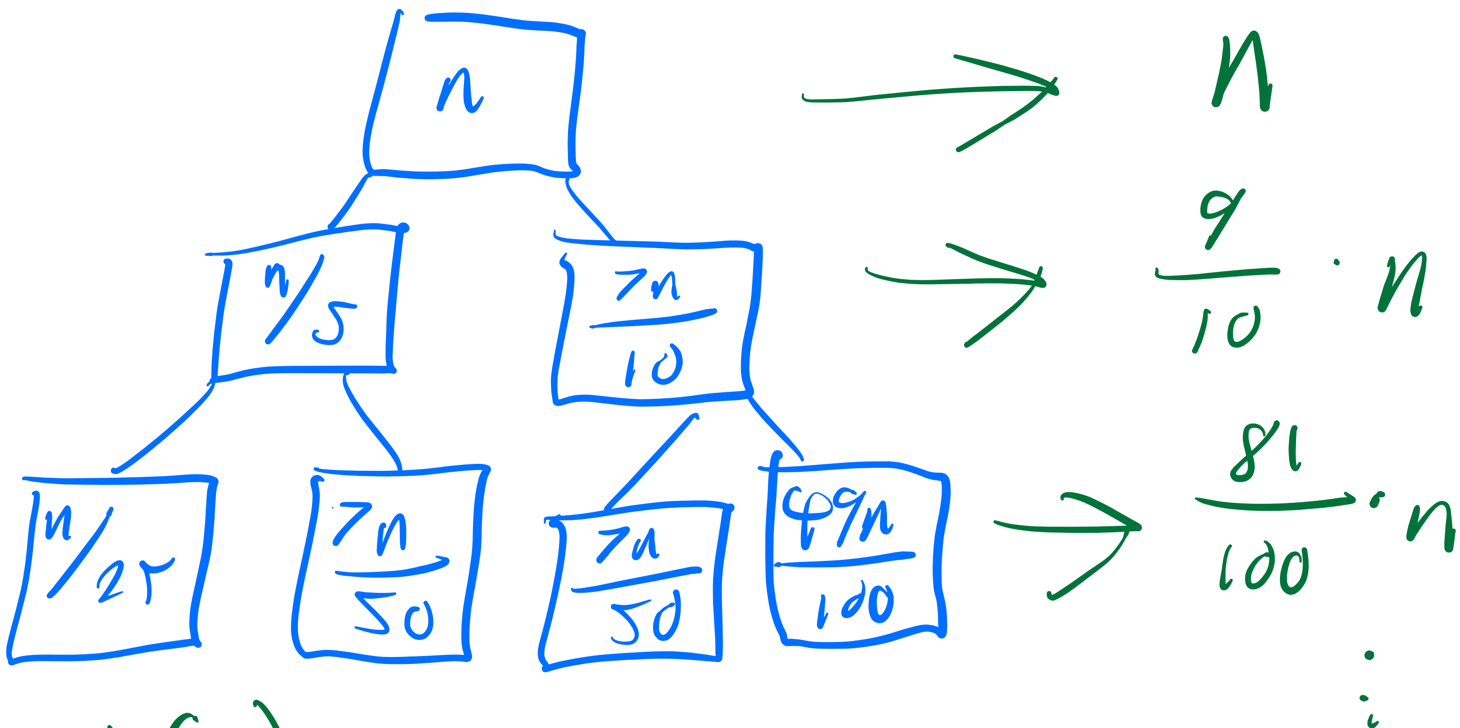
sort each column

sort columns by their medians

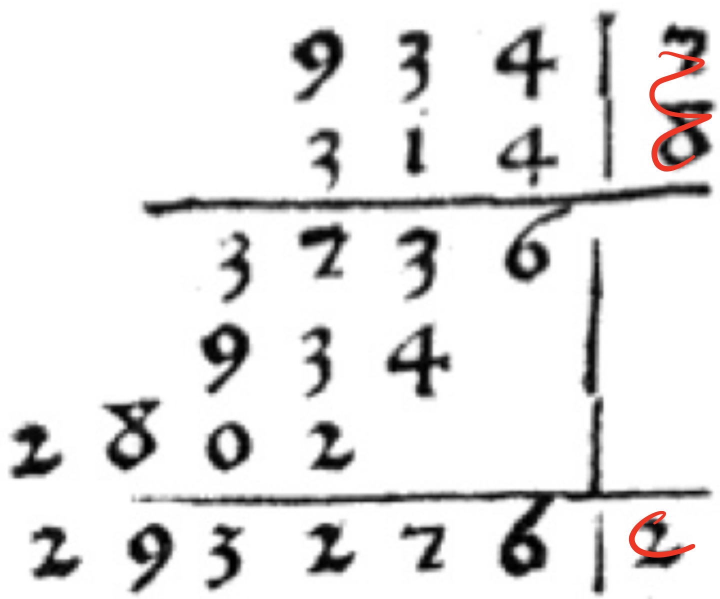
$3 \cdot \lceil \lceil n/5 \rceil / 2 \rceil \geq 3n/10$ that are smaller than mom

$\Rightarrow \leq \frac{7n}{10}$ elements bigger than mom by symmetry $\le \frac{7n}{10}$ smaller than mom

$$T(n) = O(n) + T(n/5) + T(7n/10)$$



$$T(n) = O(n)$$



$O(n^2)$ for two n-digit numbers

We have n-digit non-negative integers x & y.

Let $m = \lceil n/2 \rceil$.

$n-m$ digits, m digits

$$x = a \cdot 10^m + b$$

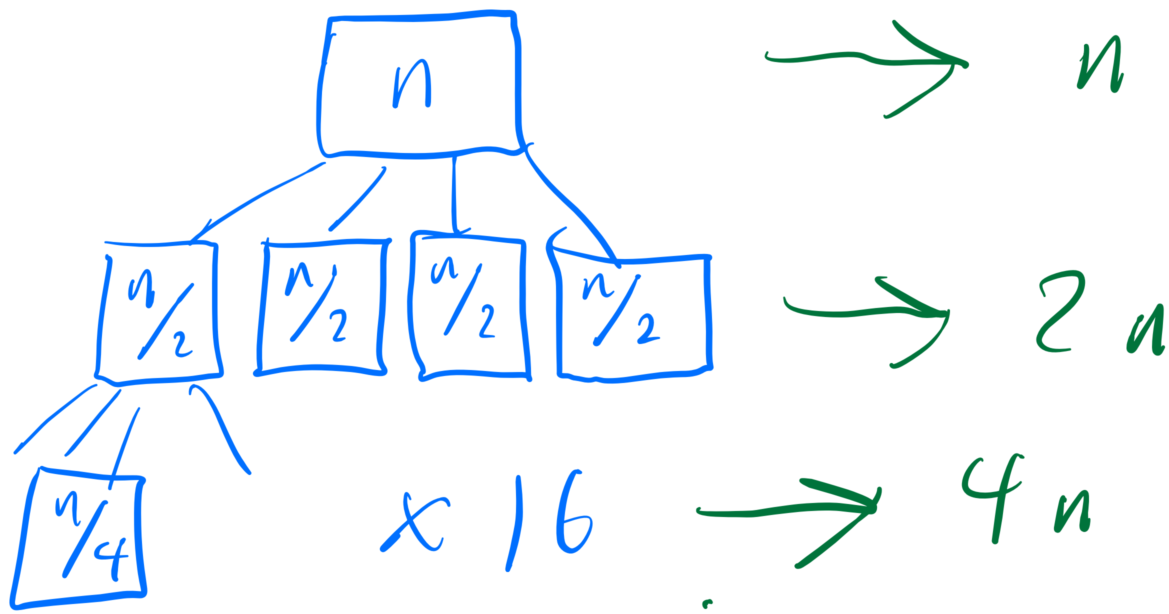
$$y = c \cdot 10^m + d$$

$$xy = (a \cdot 10^m + b)(c \cdot 10^m + d) = ac \cdot 10^{2m} + ad \cdot 10^m$$

$+\, bc \cdot 10^m + bd$

```
SplitMultiply(x, y, n):
  if n = 1
      return x · y
  else
      m ← ⌈n/2⌉
      a ← ⌊x/10^m⌋;  b ← x mod 10^m        ⟨⟨x = 10^m a + b⟩⟩
      c ← ⌊y/10^m⌋;  d ← y mod 10^m        ⟨⟨y = 10^m c + d⟩⟩
      e ← SplitMultiply(a, c, m)
      f ← SplitMultiply(b, d, m)
      g ← SplitMultiply(b, c, m)
      h ← SplitMultiply(a, d, m)
      return 10^{2m} e + 10^m (g + h) + f
```

$$T(n) = 4\,T(n/2) + O(n)$$

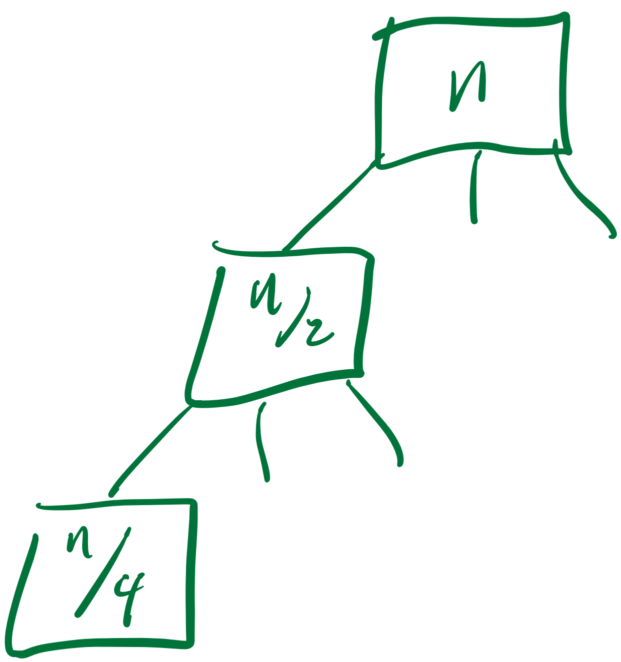


$$T(n) = O(2^{depth} \cdot n) = O(2^{\log_2 n} \cdot n) = O(n^2)$$

But! $bc + ad = ac + bd - (a-b)(c-d)$

<u>FASTMULTIPLY(x, y, n):</u>
  if $n = 1$
    return $x \cdot y$
  else
    $m \leftarrow \lceil n/2 \rceil$
    $a \leftarrow \lfloor x/10^m \rfloor$; $b \leftarrow x \bmod 10^m$ ⟨⟨$x = 10^m a + b$⟩⟩
    $c \leftarrow \lfloor y/10^m \rfloor$; $d \leftarrow y \bmod 10^m$ ⟨⟨$y = 10^m c + d$⟩⟩
    $e \leftarrow$ FASTMULTIPLY(a, c, m)
    $f \leftarrow$ FASTMULTIPLY(b, d, m)
    $g \leftarrow$ FASTMULTIPLY$(a - b, c - d, m)$
    return $10^{2m} e + 10^m (e + f - g) + f$

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



$\rightarrow n$

$\rightarrow \frac{3}{2} n$

$\rightarrow \frac{9}{4} n$

$$T(n) = O\left(\left(\frac{3}{2}\right)^{\log_2 n} \cdot n\right)$$

$$= O\left(n^{\log_2 3/2} \cdot n\right)$$

$$= O\left(n^{(\log_2 3) - 1} \cdot n\right)$$

$$= O\left(n^{\log_2 3}\right)$$

$$= O\left(n^{1.59}\right)$$

2019: $O(n \log n)$