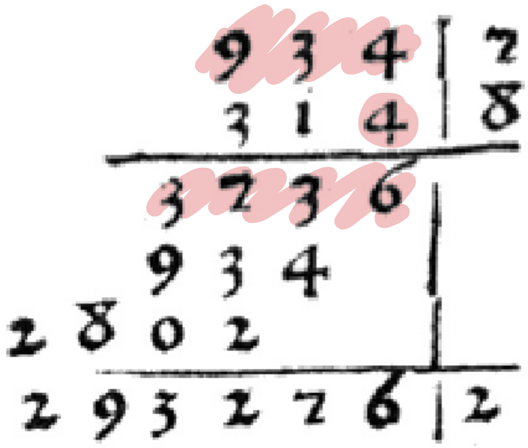
Algorithm = αλγος + αριθμος
algos arithmos
pain number

Al-Khwarizmi



$O(n^2)$ time

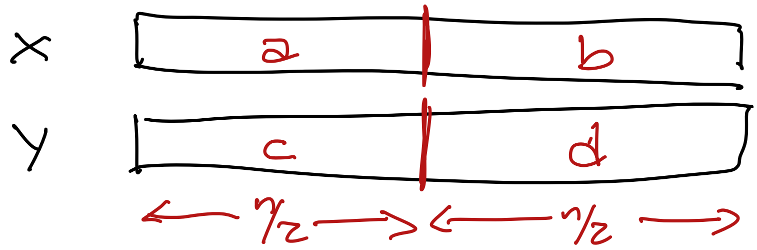
```
PEASANTMULTIPLY(x, y):
  prod ← 0
  while x > 0
      if x is odd
          prod ← prod + y
      x ← ⌊x/2⌋
      y ← y + y
  return prod
```

x	y		*prod*
			0
123	+ 456	=	456
61	+ 912	=	1368
30	~~1824~~		
15	+ 3648	=	5016
7	+ 7296	=	12312
3	+ 14592	=	26904
1	+ 29184	=	**56088**

n digits $\Rightarrow$ $O(n^2)$ time

Kolmogorov : n^2-conjecture

Karatsuba : Nuh-uh.



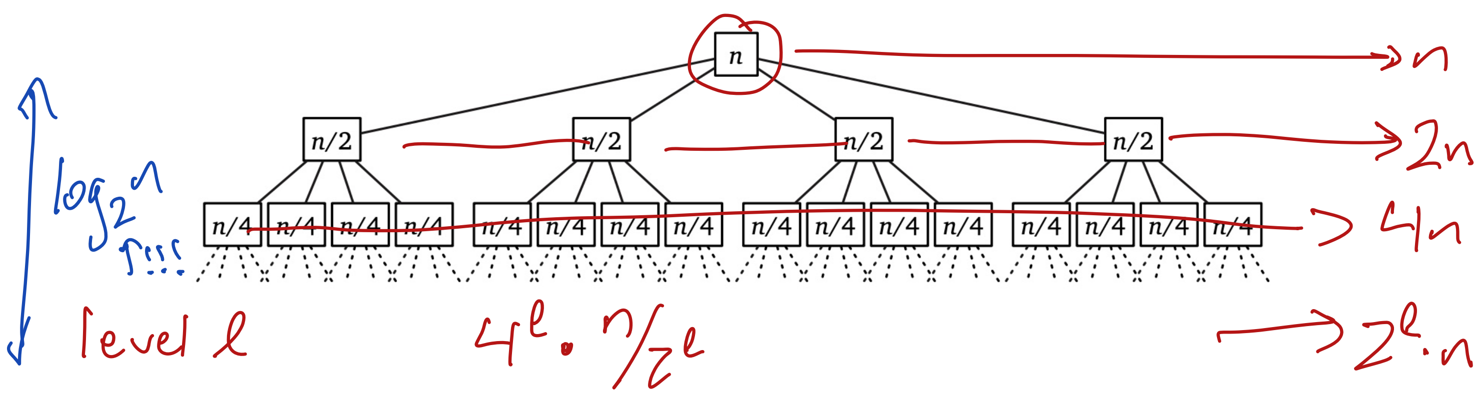
$x = a \cdot 10^{n/2} + b$

$y = c \cdot 10^{n/2} + d$

$$x \cdot y = ac \cdot 10^{n} + (bc + ad) \cdot 10^{n/2} + bd$$

```
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
```

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



Increasing geom. series — Only largest term matters

$2^{\log_2 n} \cdot n = n^2$

$\#\text{leaves} = 4^{\log_2 n} = 2^{2\log_2 n} = n^2$

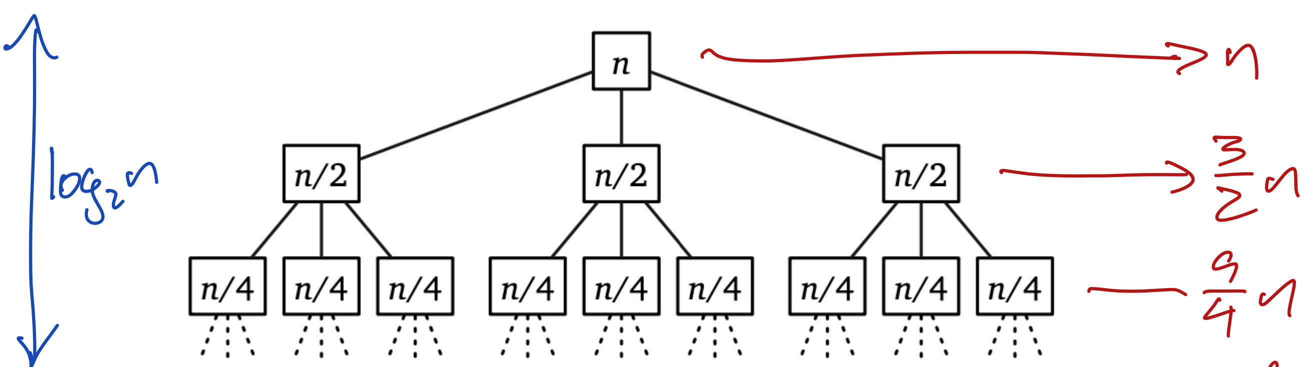
$$x \cdot y = ac \cdot 10^{n} + (bc + ad) \cdot 10^{n/2} + bd$$

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

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

```
FastMultiply(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 ← FastMultiply(a, c, m)
        f ← FastMultiply(b, d, m)
        g ← FastMultiply(a − b, c − d, m)
        return 10^{2m} e + 10^m (e + f − g) + f
```

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



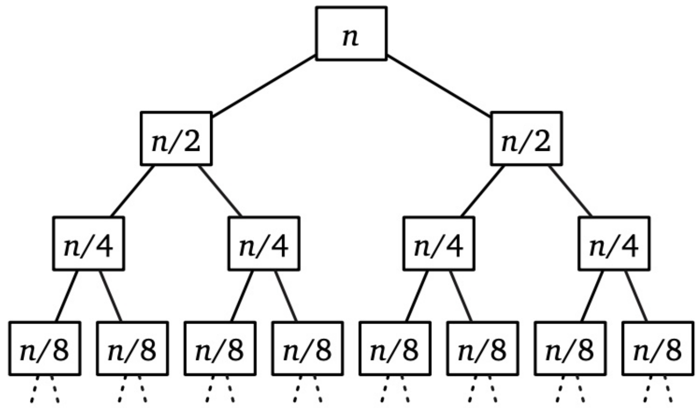
level ℓ: $\quad 3^{\ell} \cdot n/2^{\ell} \quad\quad \left(\frac{3}{2}\right)^{\ell} \cdot n$

$$\# \text{leaves} = 3^{\log_2 n} = n^{\log_2 3}$$

$$O(n^{\log_2 3}) \approx O(n^{1.6\ldots})$$

2019: $O(n \log n)$ time!

```
MergeSort(A[1..n]):
  if n > 1
      m ← ⌊n/2⌋
      MergeSort(A[1..m])        ⟨⟨Recurse!⟩⟩
      MergeSort(A[m+1..n])      ⟨⟨Recurse!⟩⟩
      Merge(A[1..n], m)
```

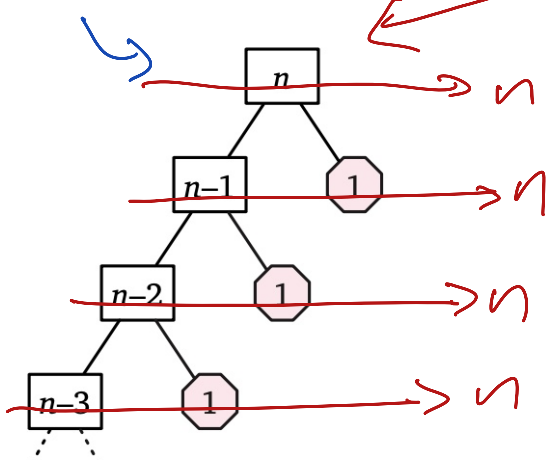


$$T(n) = 2T(n/2) + O(n)$$

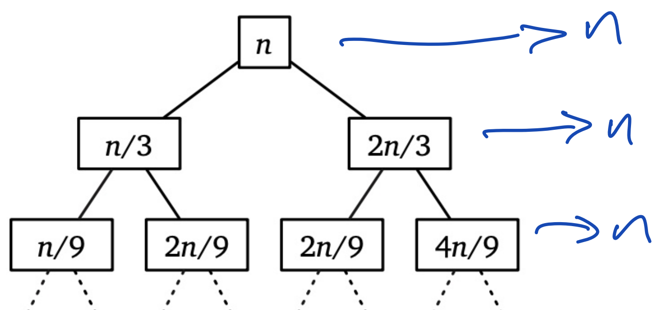
```
QuickSort(A[1..n]):
  if (n > 1)
      Choose a pivot element A[p]
      r ← Partition(A, p)
      QuickSort(A[1..r−1])      ⟨⟨Recurse!⟩⟩
      QuickSort(A[r+1..n])      ⟨⟨Recurse!⟩⟩
```

$$T(n) \le \max_r \big(T(r-1) + T(n-r)\big) + O(n)$$
$$\le \cancel{T(0)} + T(n-1) + O(n)$$

Worst:



Lucky:



$$T(n) \le T(\tfrac{n}{3}) + T(\tfrac{2n}{3}) + O(n)$$

$O(n\log n)$

At most n levels
each at most n work

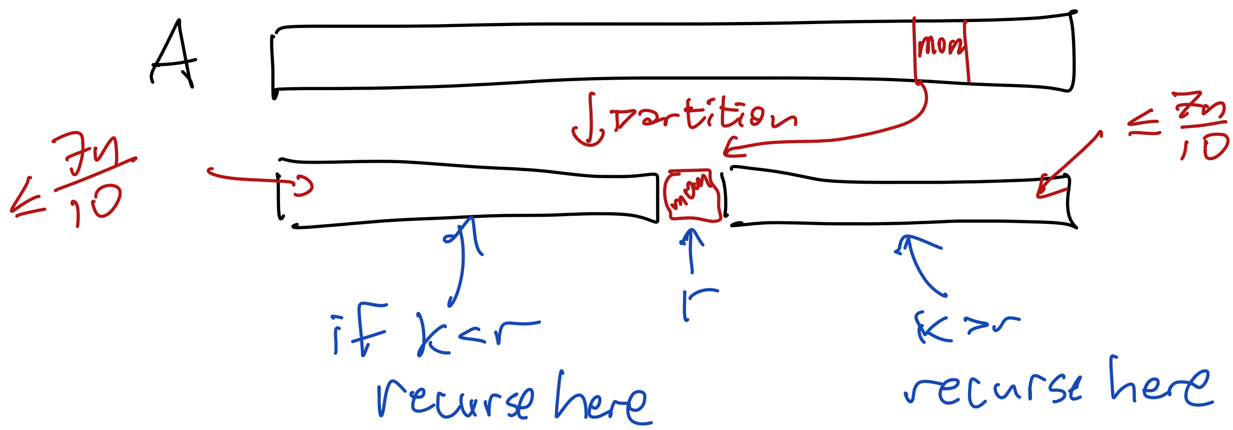
$\ge n/2$ levels $\ge \frac{n}{2}$ work

Selection: Given unsorted array $A[1..n]$
integer k

Find kth smallest element of A.

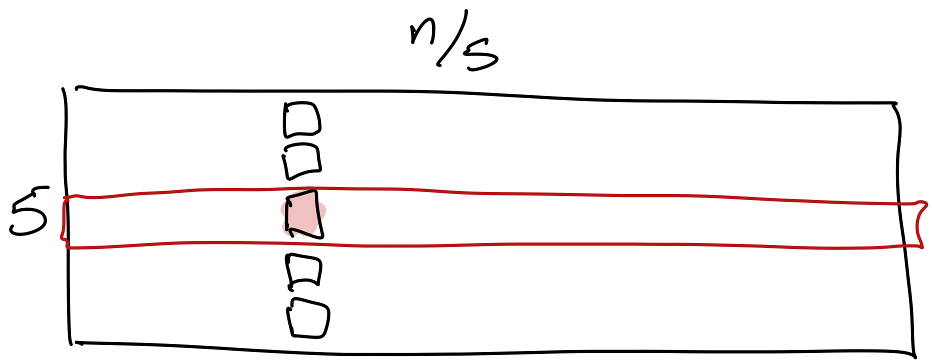
"One-Armed Quicksort"

```
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]
```



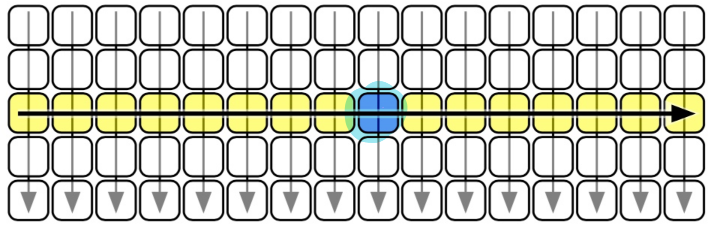
$$T(n) \le \max\{T(r-1), T(n-r)\} + O(n)$$
$$\le T(n-1) + O(n) = \underline{\underline{O(n^2)}}$$

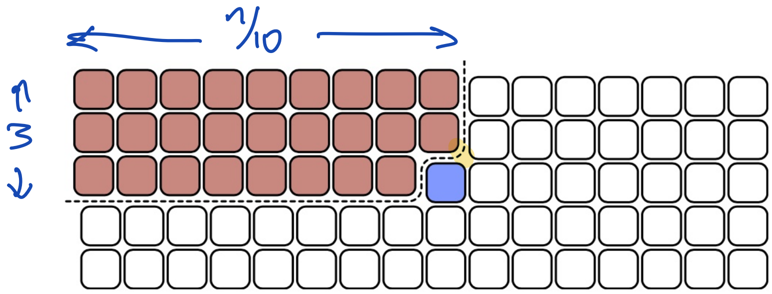
Blum
Floyd
Pratt
Rivest
Tarjan



O(n): Find median of each chunk of 5

Compute median of medians — RECURSE?





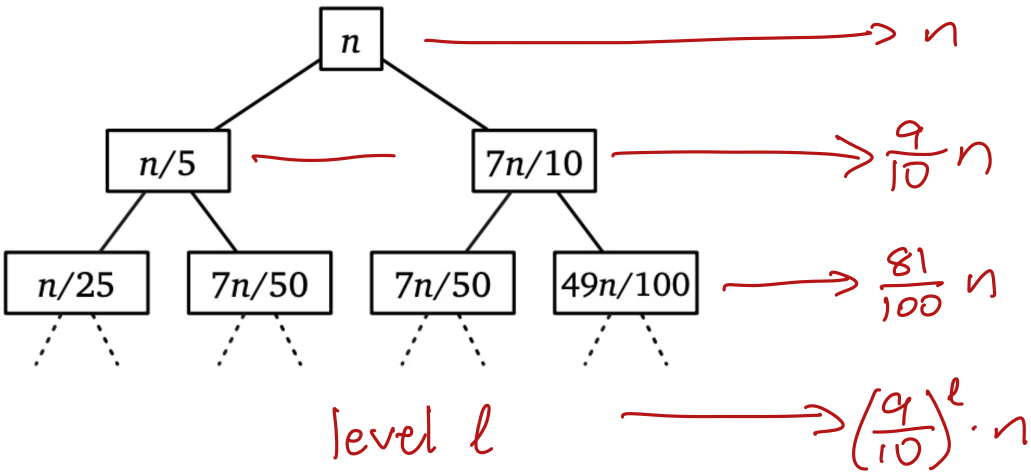
mom is bigger than $\geq \frac{n}{10} \times 3$ elements of A

$\Rightarrow$ mom $\leq \frac{7n}{10}$ elements of A

MomSelect:

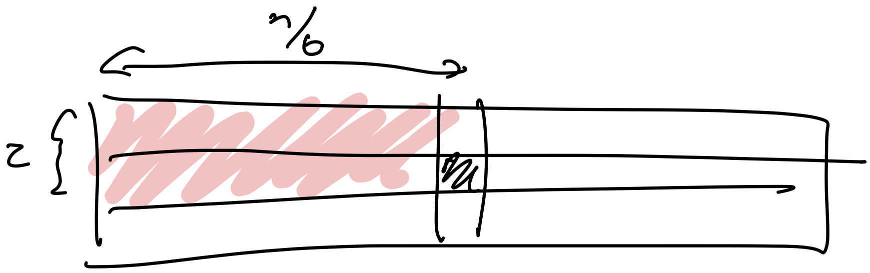
$$T(n) \leq O(n) + T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right)$$

```
MomSelect(A[1..n], k):
  if n ≤ 25   ⟨⟨or whatever⟩⟩
      use brute force
  else
      m ← ⌈n/5⌉
      for i ← 1 to m
          M[i] ← MedianOfFive(A[5i − 4..5i])   ⟨⟨Brute force!⟩⟩
      mom ← MomSelect(M[1..m], ⌊m/2⌋)        ⟨⟨Recursion!⟩⟩
      r ← Partition(A[1..n], mom)
      if k < r
          return MomSelect(A[1..r − 1], k)    ⟨⟨Recursion!⟩⟩
      else if k > r
          return MomSelect(A[r + 1..n], k − r)    ⟨⟨Recursion!⟩⟩
      else
          return mom
```



O(n) time!

16



$T(n) \in O(n) + O(\frac{n}{3}) + O(\frac{2n}{3})$