Partition A into $<A[p], A[p], >A[p]$
return new index of $A[p]$

```
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!⟩⟩
```

```
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	P	L
Partition:	A	G	O	E	I	N	L	M	P	T	X	S	R
Recurse Left:	A	E	G	I	L	M	N	O	P	T	X	S	R
Recurse Right:	A	E	G	I	L	M	N	O	P	R	S	T	X

p=12
r=9

$$T(n) \leq O(n) + \max_r \left(T(r-1) + T(n-r)\right)$$

$r=1$ or $r=n \Rightarrow T(n) \leq O(n) + T(n-1)$

$T(n) \leq O(n^2)$

Magic $\Rightarrow r = n/2$

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

$= O(n \log n)$

ASSUME

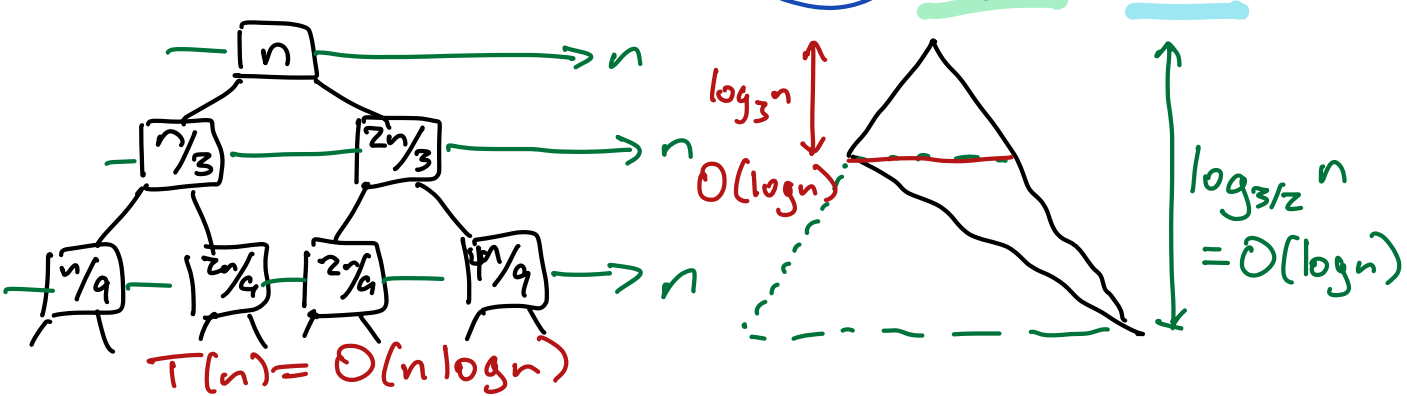
Less magic $\Rightarrow \frac{n}{3} \leq r \leq \frac{2n}{3}$

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

$\log_b n = \frac{\log n}{\log b}$

max at $r = \frac{n}{3}$ or $r = \frac{2n}{3}$

$$T(n) = O(n) + T\left(\tfrac{n}{3}\right) + T\left(\tfrac{2n}{3}\right)$$

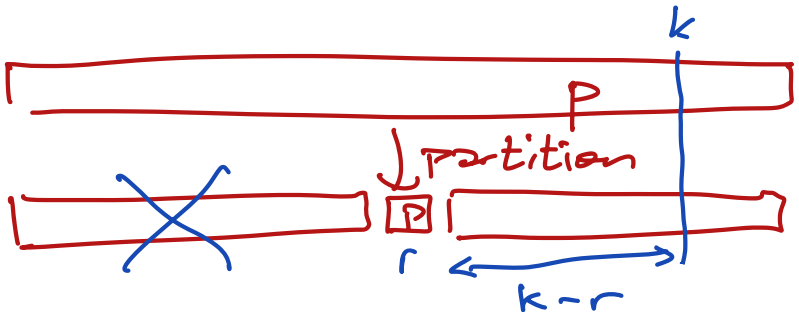


$O(n)$ = anonymous function $F(n)$

s.t. $\lim_{n \to \infty} \frac{F(n)}{n}$ is finite

return k^{th} smallest in A

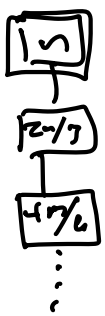
```
QuickSelect(A[1..n], k):
  if n = 1
      return A[1]
  else
      Choose a pivot element A[p]
      r ← Partition(A[1..n], p)
      if k < r
          return QuickSelect(A[1..r−1], k)
      else if k > r
          return QuickSelect(A[r+1..n], k−r)
      else
          return A[r]
```



$$T(n) \le O(n) + \max(\max(T(r-1), T(n-r)))$$

$$\le O(n) + T(n-1) = O(n^2)$$

Magic: $\frac{n}{3} \le r \le \frac{2n}{3} \Rightarrow T(n) \le O(n) + T(\frac{2n}{3}) = O(n)$




```
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
```

Blum Floyd Pratt Rivest Tarjan 1970s

7	56	21	91	62	70	13	50	38	60	31	22	25	11	36	78	53	77	35	29
24	28	48	80	44	4	20	10	64	45	49	65	6	94	68	26	72	95	75	47
88	82	52	5	30	41	16	85	14	19	40	43	18	90	81	67	73	17	39	51
23	3	79	37	86	34	76	96	63	93	8	15	83	46	97	12	87	69	2	84
74	42	1	66	9	58	57	61	33	92	71	89	0	98	32	27	59	54	99	55

↓ O(1) time

7	3	1	5	9	4	13	10	14	19	8	15	0	11	32	12	53	17	2	29
23	28	21	37	30	34	16	50	33	45	31	22	6	46	36	26	59	54	35	47
24	42	48	66	44	41	20	61	38	60	40	43	18	90	68	27	72	69	39	51
74	56	52	80	62	58	57	85	63	92	49	65	25	94	81	67	73	77	75	55
88	82	79	91	86	70	76	96	64	93	71	89	83	98	97	78	87	95	99	84

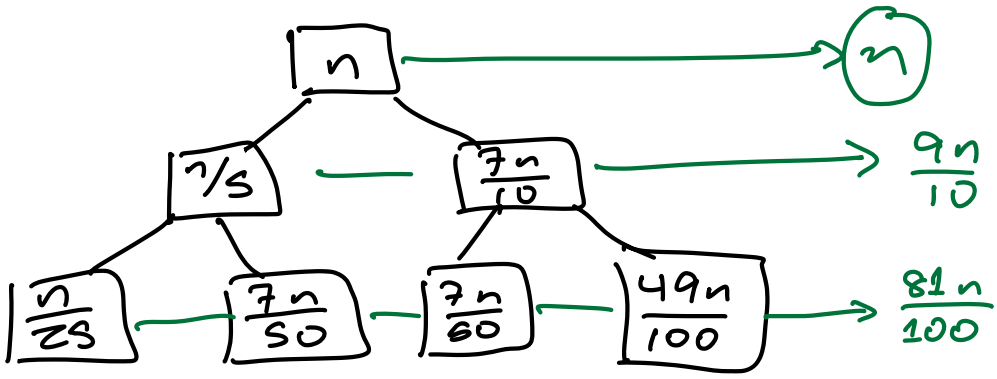
7	3	1	5	9	4	13	10	14	19	8	15	0	11	32	12	53	17	2	29
23	28	21	37	30	34	16	50	33	45	31	22	6	46	36	26	59	54	35	47
24	42	48	66	**44**	41	20	61	38	60	40	43	18	90	68	27	72	69	39	51
74	56	52	80	62	58	57	85	63	92	49	65	25	94	81	67	73	77	75	55
88	82	79	91	86	70	76	96	64	93	71	89	83	98	97	78	87	95	99	84

7	3	1	5	9	4	13	10	14	19	8	15	0	11	32	12	53	17	2	29
23	28	21	37	30	34	16	50	33	45	31	22	6	46	36	26	59	54	35	47
24	42	48	66	**44**	41	20	61	38	60	40	43	18	90	68	27	72	69	39	51
74	56	52	80	62	58	57	85	63	92	49	65	25	94	81	67	73	77	75	55
88	82	79	91	86	70	76	96	64	93	71	89	83	98	97	78	87	95	99	84

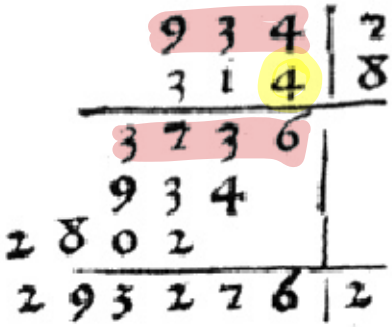
8	15	0	7	3	4	13	14	12	2	9	53	17	10	19	11	32	29	1	5
31	22	6	23	28	34	16	33	26	35	30	59	54	50	45	46	36	47	21	37
40	43	18	24	42	41	20	38	27	39	**44**	72	69	61	60	90	68	51	48	66
49	65	25	74	56	58	57	63	67	75	62	73	77	85	92	94	81	55	52	80
71	89	83	88	82	70	76	64	78	99	86	87	95	96	93	98	97	84	79	91

3n/10 mom 13n/10

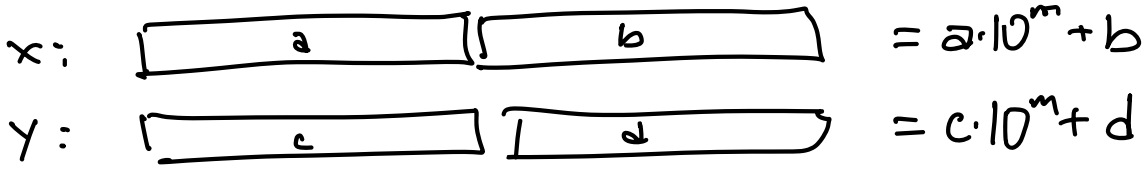
$$T(n) \leq O(n) + T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) = \boxed{O(n)}$$



descending geom. series



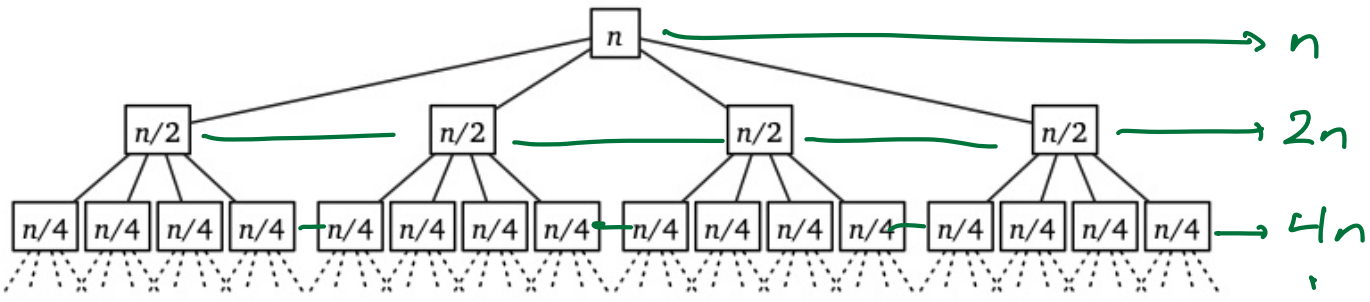
$O(n^2)$ time



$$xy = ac \cdot 10^{2m} + ad\,10^m + bc\,10^m + bd$$

$$T(n) = O(n) + 4T(\tfrac{n}{2})$$

```
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
```



$\ell = \log_2 n$

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

$x = 10^m a + b$
$y = 10^m c + d$

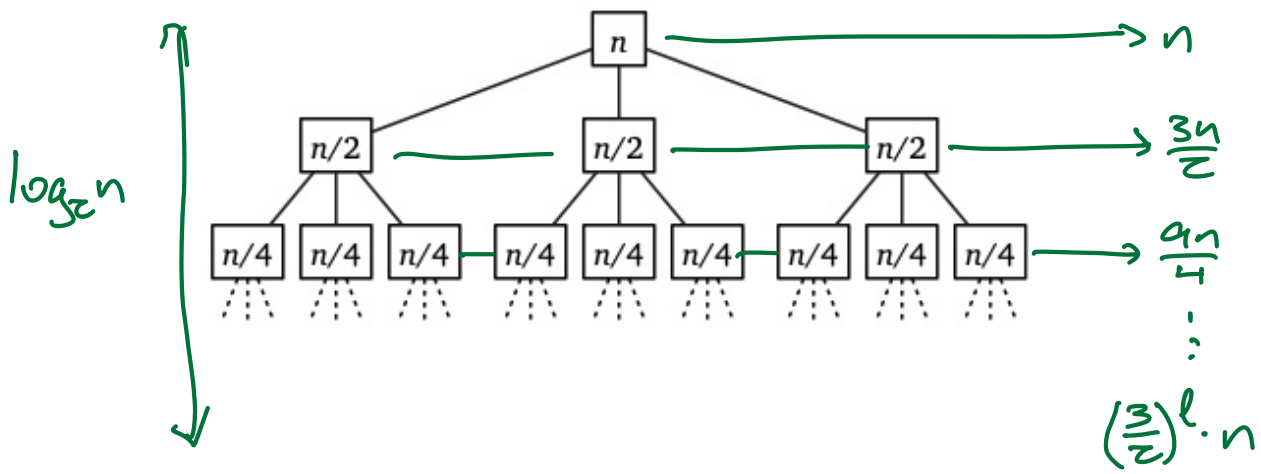
ac $\quad$ bd

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

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

```
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
```

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



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

$= n^{\log_2 3 - 1} \cdot n$

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

$= O(n^{1.61\ldots})$

$\boxed{a^{\log_b c} = c^{\log_b a}}$

$= e^{\frac{\ln a \cdot \ln c}{\ln b}}$

$O(n \log n)$ 2019