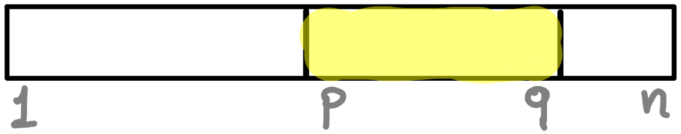
RANDSELECT (randomized Selection)

Given n unsorted elements in an array (or linked list)

find the one with rank r → r-th smallest.

- For simplicity, assume no duplicates → Easy to handle.
- If necessary shuffle data to make random order.

Recursive function: RandSelect(k, p, q)

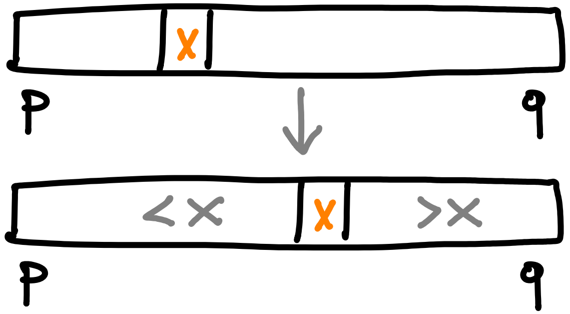


returns k-th smallest in subarray from index p to index q.

We start with RandSelect$(r, 1, n)$

// Find k-th smallest within [p,q]

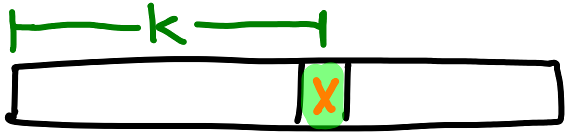
RandSelect(k, p, q)



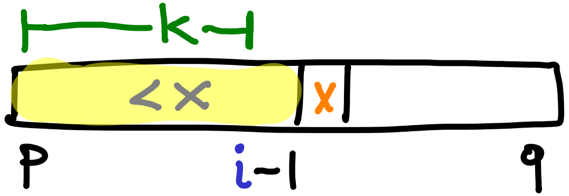
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]   = 1 + #elements smaller than X

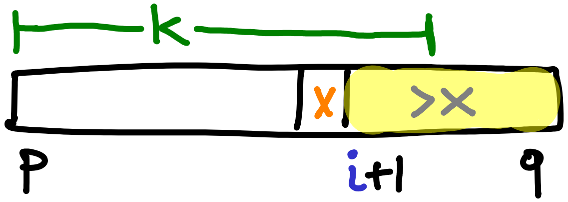
3) if rank(x) = k, return x



if k < rank(x), RandSelect(k, p, i−1)



if rank(x) < k, RandSelect(k − rank(x), i+1, q)



Example: Find 7th smallest

$k = r = 7, \quad p = 1, \quad q = n = 12$

11	10	8	13	9	3	2	6	5	1	12	15
1	2	3	4	5	6	7	8	9	10	11	12

RandSelect(7, 1, 12) →

5	10	8	1	9	3	2	6	11	13	12	15
1	2	3	4	5	6	7	8	9	10	11	12

↳ Partition: pivot = x = 11

$7 = k < \mathrm{Rank}(11) = 9$

RandSelect(7, 1, 8) →

1	2	3	5	9	8	10	6	11	13	12	15
1	2	3	4	5	6	7	8	9	10	11	12

↳ Partition: x = 5

$4 = \mathrm{Rank}(5) < k = 7$

RandSelect(3, 5, 8) →

1	2	3	5	6	8	9	10	11	13	12	15
1	2	3	4	5	6	7	8	9	10	11	12

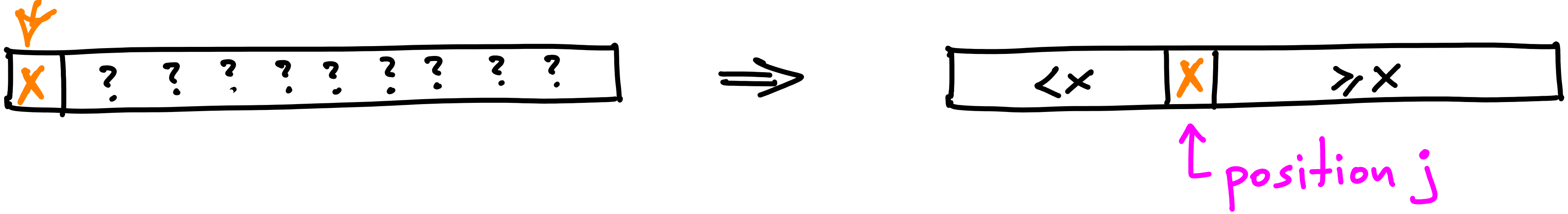
↳ Partition: x = 9

return 9

$\mathrm{Rank}(9) = 3$

RANDSELECT recap

- If necessary shuffle data to make random order.
- choose a pivot & partition. $\rightarrow \Theta(n)$



- in the worst case, RandSelect the larger side.

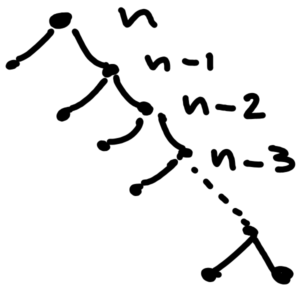
$$T(n) = \Theta(n) + \max\{T(j-1), T(n-j)\}$$

$$T(n) = \Theta(n) + \max\{T(j-1), T(n-j)\}$$

What is the worst-case time complexity, and why?

$\hookrightarrow$ already sorted input, reverse-sorted, nearly sorted...

$$T(n) = T(n-1) + \Theta(n) \quad = \Theta(n^2)$$



What would be ideal? (assuming we must actually recurse)

$\hookrightarrow$ ~ balanced partition, every time

$$T(n) = T\left(\frac{n}{2}\right) + \Theta(n) \quad = \Theta(n)$$

What if we always have a "sort-of-balanced" partition?

e.g., $T(n) = T\left(\frac{9n}{10}\right) + \Theta(n) \quad = \Theta(n)$

Expected time: call a split balanced if pivot ranks in $\left[\frac{n}{4} \dots \frac{3n}{4}\right]$

unbalanced otherwise

Worst case if balanced split: $T(n) \leq T\left(\frac{3n}{4}\right) + dn$

Worst case if unbalanced split: $T(n) \leq T(n-1) + dn < T(n) + dn$

Each split has a 50% chance of being balanced

$$T(n) \leq 0.5\left(T(n) + dn\right) + 0.5 \cdot \left(T\left(\frac{3n}{4}\right) + dn\right)$$

$$0.5\,T(n) \leq dn + 0.5 \cdot T\left(\frac{3n}{4}\right)$$

$$T(n) \leq T\left(\frac{3n}{4}\right) + 2dn = \Theta(n)$$

$$2dn \cdot \frac{1}{1 - 3/4} = 8dn$$