RandSelect (randomized Selection)

RandSelect (randomized Selection)

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

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

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.

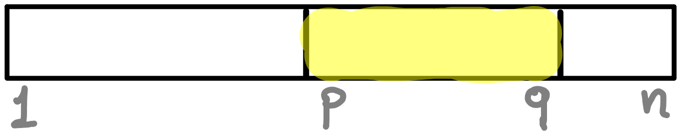
RANDSELECT (randomized Selection)

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)

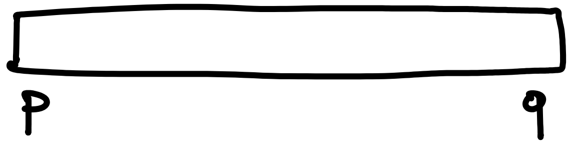


returns k-th smallest in subarray from index p to index q.

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

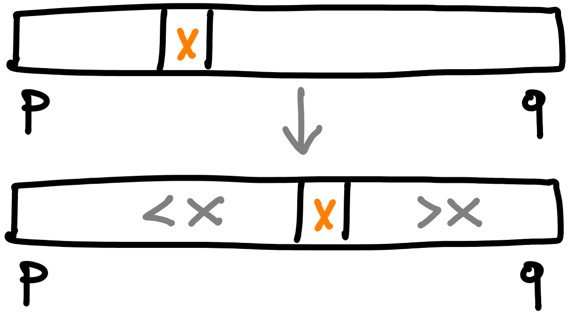
// Find k-th smallest within [p,q]

RandSelect(k, p, q)



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

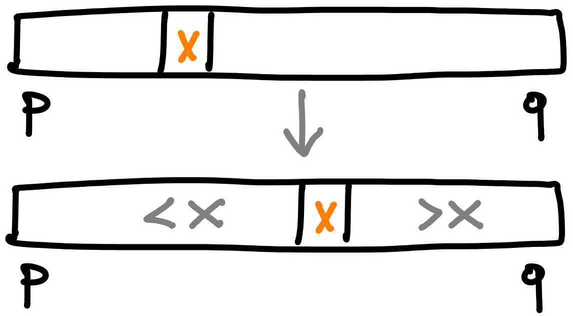
RandSelect(k, p, q)



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

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

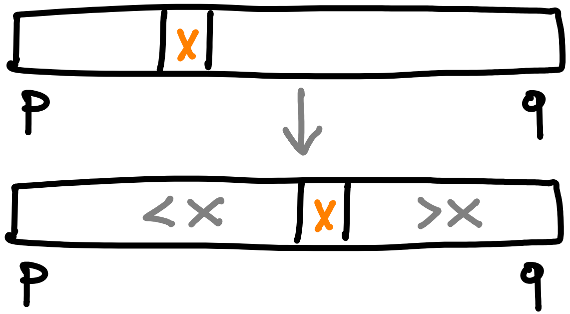
RandSelect(k, p, q)



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

2) Calculate rank of x within [p,q]

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



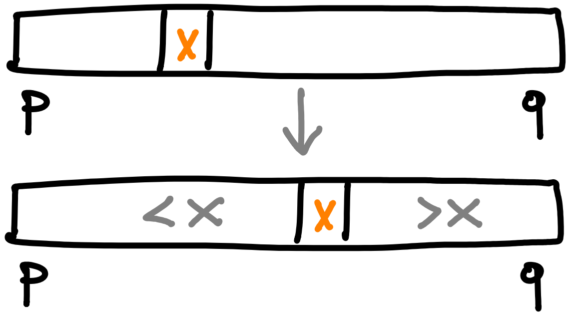
RandSelect(k, p, q)

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

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

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

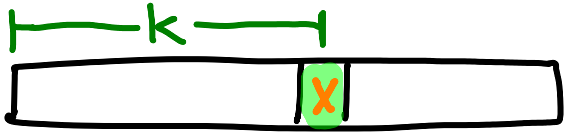
RandSelect(k, p, q)



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

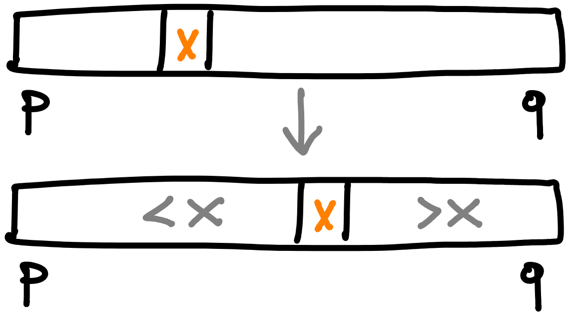
2) Calculate rank of x within [p,q]

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



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

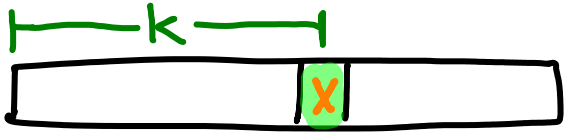
RandSelect(k, p, q)



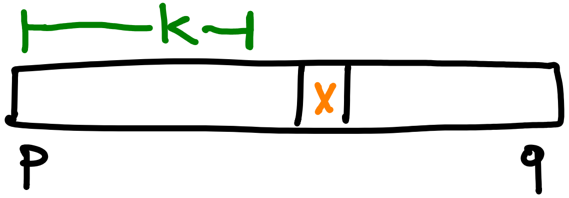
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

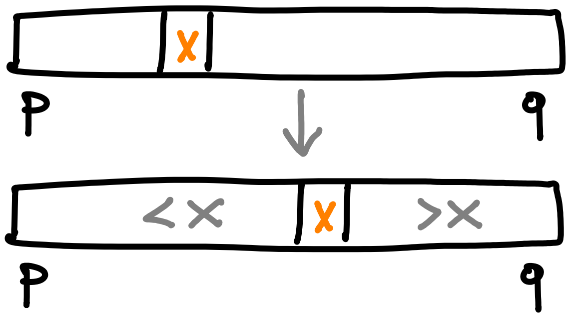
3) if rank(x) = k, return x



if k < rank(x) ... ?



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

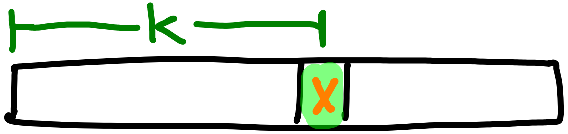


RandSelect(k, p, q)

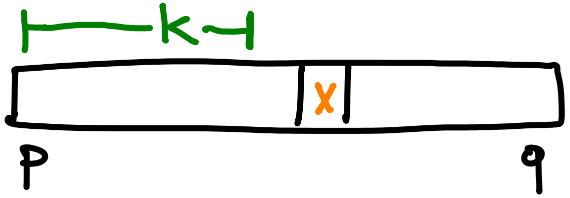
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

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

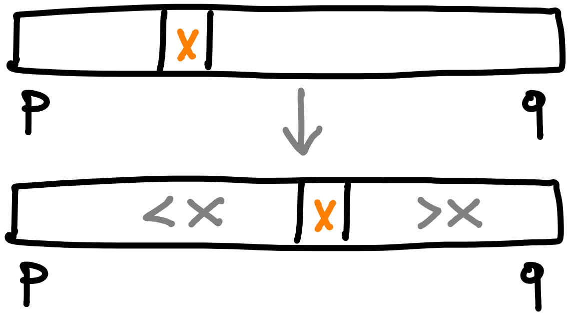


if k < rank(x), RandSelect(?, ?, ?)



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

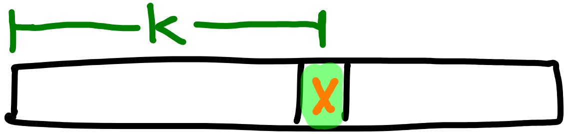
RandSelect(k, p, q)



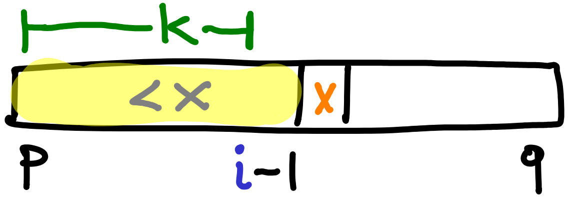
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

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

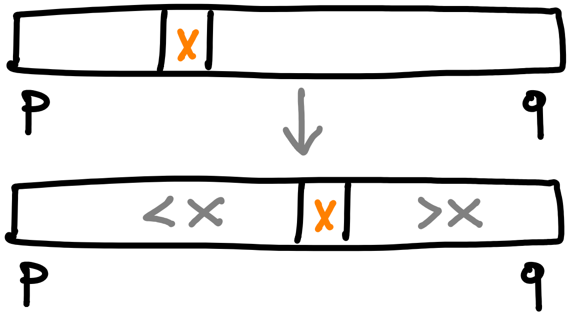


if k < rank(x), RandSelect(?, ?, ?)



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

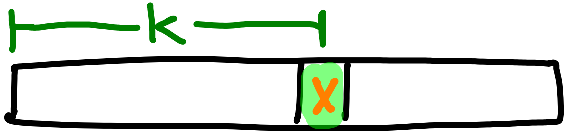
RandSelect(k, p, q)



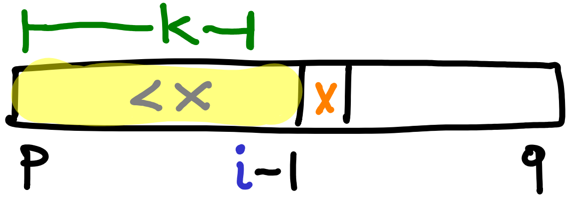
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

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

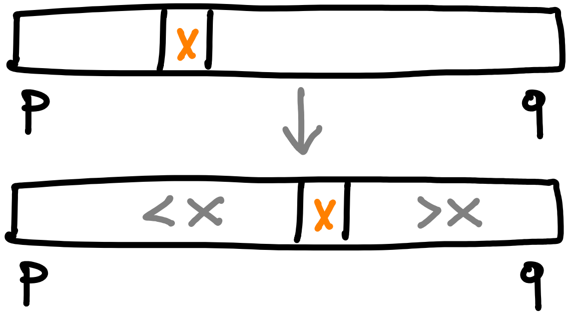


if k < rank(x), RandSelect(k, p, i−1)



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

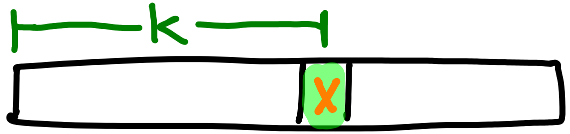
RandSelect(k, p, q)



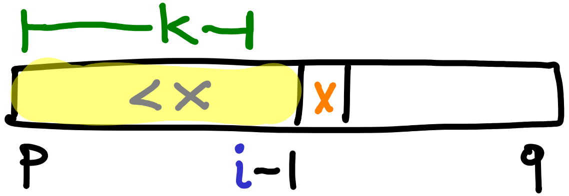
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

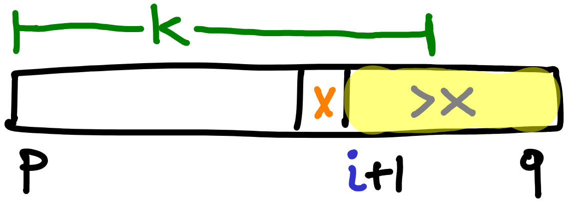
3) if rank(x) = k, return x



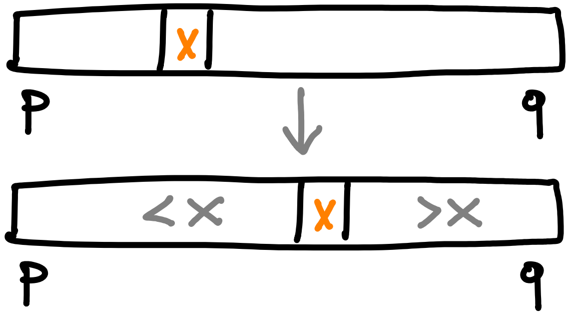
if $k < \text{rank}(x)$, RandSelect(k, p, i−1)



if $\text{rank}(x) < k$, RandSelect(?, ?, ?)



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

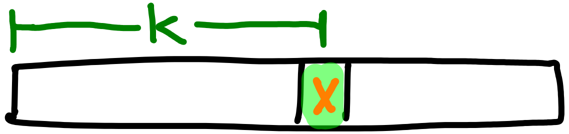


RandSelect(k, p, q)

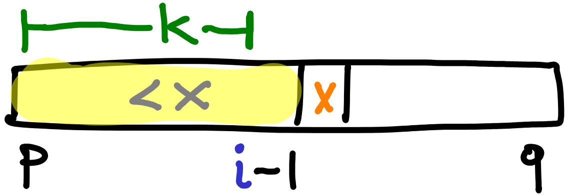
1) Use a random pivot x to partition [p,q]

2) Calculate rank of x within [p,q]

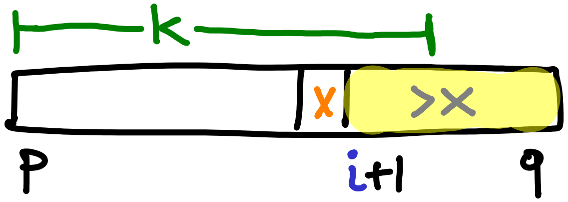
3) if rank(x) = k, return x



if k < rank(x), RandSelect(k, p, i−1)



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



Example: Find 7th smallest

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

Example: Find 7th smallest

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

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

RandSelect(7, 1, 12)

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)

↳ Partition: pivot $= x = 11$

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) →

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

↳ Partition: pivot $= x = 11$

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) →

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

↳ Partition: pivot $= x = 11$

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) →

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

↳ Partition: pivot = x = 11

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 $= x = 11$

Example: Find 7th smallest

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

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 $= x = 11$

Rank(11) = 9

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 $= x = 11$

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

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 $= x = 11$

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

RandSelect(7, 1, 8)

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 $= x = 11$

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

RandSelect$(7, 1, 8)$

↳ Partition: $x = 5$

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 = $x = 11$

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

RandSelect(7, 1, 8) →

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: $x = 5$

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) →

↳ Partition: pivot $= x = 11$

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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) $\rightarrow$

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 $= x = 11$

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

RandSelect(7, 1, 8) $\rightarrow$

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

↳ Partition: $x = 5$

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 $= x = 11$

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

RandSelect(7, 1, 8) →

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

↳ Partition: $x = 5$

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) →

↳ Partition: pivot $= x = 11$

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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) $\rightarrow$

↳ Partition: pivot = x = 11

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

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

RandSelect(7, 1, 8) $\rightarrow$

↳ Partition: $x = 5$

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

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) →

↳ Partition: pivot = x = 11

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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) →

↳ Partition: pivot $= x = 11$

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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

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

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

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

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

RandSelect(3, 5, 8)

$k - \text{Rank}(5) = 3$

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 $= x = 11$

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

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: $x = 5$

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

RandSelect(3, 5, 8)

↳ Partition: $x = 9$

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

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

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: $x = 5$

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

RandSelect(3, 5, 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: $x = 9$

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

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

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: $x = 5$

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

RandSelect(3, 5, 8) →

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

↳ Partition: $x = 9$

Example: Find 7th smallest

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

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

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 = $x = 11$

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

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: $x = 5$

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

RandSelect(3, 5, 8) →

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

↳ Partition: $x = 9$

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) →

↳ Partition: pivot = x = 11

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

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

RandSelect(7, 1, 8) →

↳ Partition: x = 5

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

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

RandSelect(3, 5, 8) →

↳ Partition: x = 9

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

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) →

↳ Partition: pivot $= x = 11$

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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

RandSelect(3, 5, 8) →

↳ Partition: $x = 9$

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

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

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

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

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: $x = 5$

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

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: $x = 9$

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

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) →

↳ Partition: pivot $= x = 11$

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

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

RandSelect(7, 1, 8) →

↳ Partition: $x = 5$

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

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

RandSelect(3, 5, 8) →

↳ Partition: $x = 9$

return 9

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

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

RandSelect recap

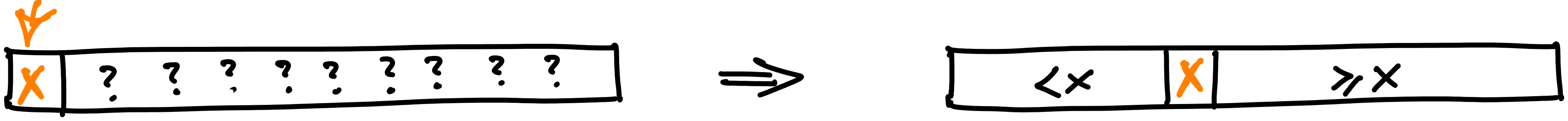
RANDSELECT recap

- If necessary shuffle data to make random order.

RANDSELECT recap

- If necessary shuffle data to make random order.

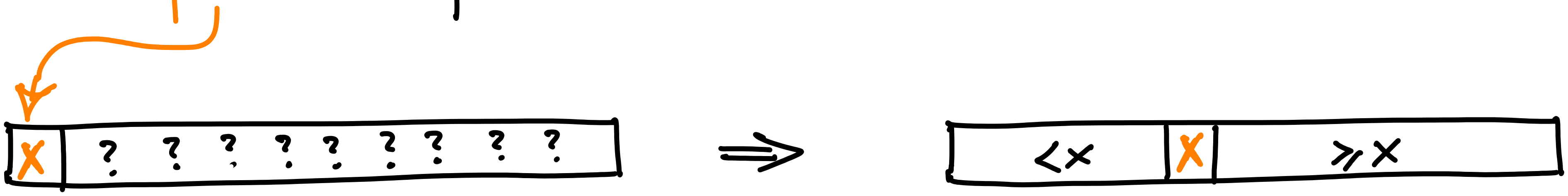
- choose a pivot & partition.



RandSelect recap

- If necessary shuffle data to make random order.

- choose a pivot & partition.

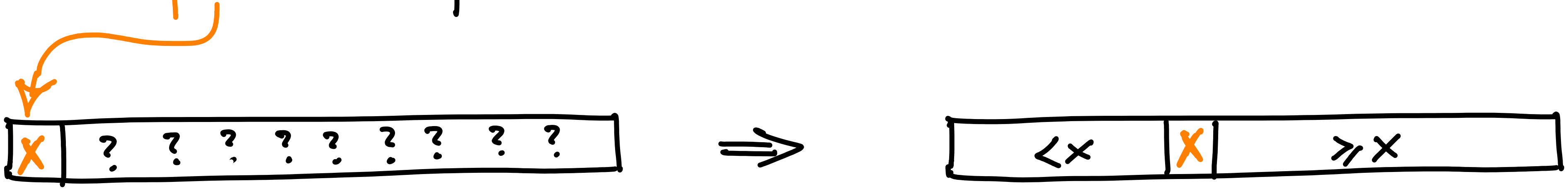


- in the worst case, RandSelect the larger side.

RANDSELECT recap

- If necessary shuffle data to make random order.

- choose a pivot & partition.



- in the worst case, RandSelect the larger side.

$T(n) = ?$

RandSelect recap

- If necessary shuffle data to make random order.

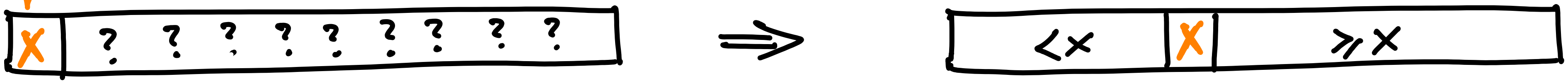
- choose a pivot & partition. $\rightarrow \underline{\Theta(n)}$

- in the worst case, RandSelect the larger side.

$$T(n) = \underline{\Theta(n)} + \ ?$$

RandSelect recap

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



position j

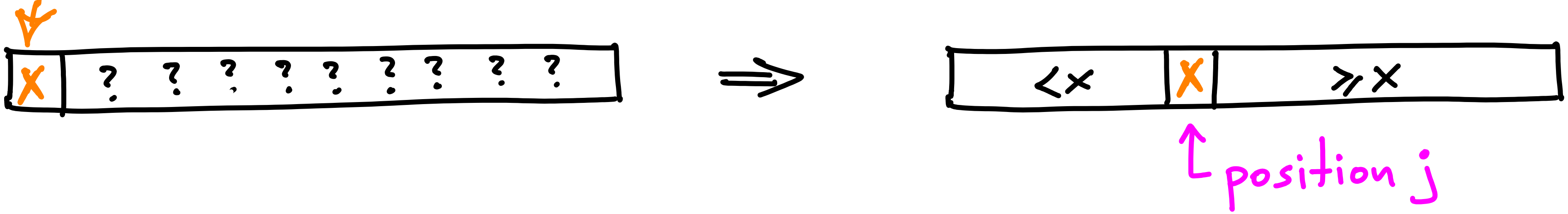
- in the worst case, RandSelect the larger side.

$$T(n) = \Theta(n) + ?$$

RandSelect recap

- If necessary shuffle data to make random order.

- choose a pivot & partition. → $\Theta(n)$



- in the worst case, RandSelect the larger side.

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

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

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

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

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

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

↳ already sorted input, reverse-sorted, nearly sorted...

$T(n) = ?$

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

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

↳ already sorted input, reverse-sorted, nearly sorted...

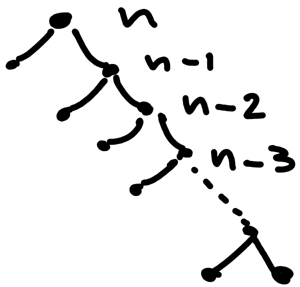
$$T(n) = T(n-1) + \Theta(n) \quad = ?$$

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

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

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

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

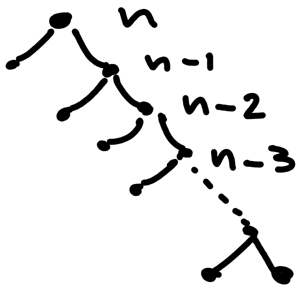


$$T(n) = \Theta(n) + \max\{T(j-1), T(n-j)\}$$

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

↳ already sorted input, reverse-sorted, nearly sorted...

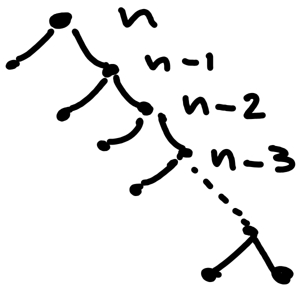
$$T(n) = T(n-1) + \Theta(n) \quad = \Theta(n^2)$$



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

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

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



↳ 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)

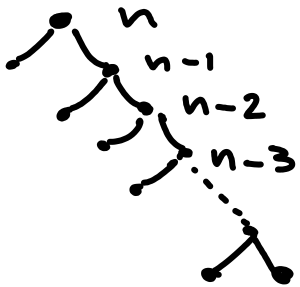
↳ ~ balanced partition, every time

$$T(n) = ?$$

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

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

↳ 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)

↳ ~ balanced partition, every time

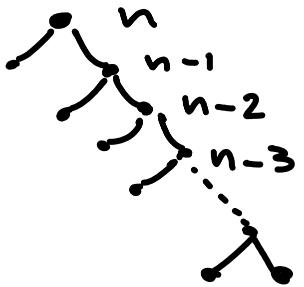
$$T(n) = T\left(\frac{n}{2}\right) + \Theta(n) \quad = \ ?$$

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

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

↳ already sorted input, reverse-sorted, nearly sorted...

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



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

↳ ~ balanced partition, every time

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

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

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

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

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

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

Worst case if balanced split: $T(n) \leq$?

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: $T(n) \leq \dots$

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

unbalanced otherwise

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

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

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

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

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

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

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

Each split has a 50% chance of being balanced

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

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

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

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

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

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

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

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

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

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

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

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

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

Expected time complexity of RandSelect

More refined calculation

always assuming worst scenario

$$T(n) \leq \Theta(n) + \text{one of}: \left.\begin{cases} T(\max\{0, n-1\}) \\ T(\max\{1, n-2\}) \\ T(\max\{2, n-3\}) \\ \vdots \\ T(\max\{n-1, 0\}) \end{cases}\right\} n \text{ possible outcomes}$$

Define $X_k \begin{cases} 1 & \text{if RandPartition gives } k \text{ vs } n-k-1 \text{ split.} \\ 0 & \text{otherwise} \end{cases}$

always assuming worst scenario

$$T(n) \leq \Theta(n) + \text{one of: } \begin{cases} T(\max\{0, n-1\}) & k = 0 \\ T(\max\{1, n-2\}) & \\ T(\max\{2, n-3\}) & \\ \vdots & \\ T(\max\{n-1, 0\}) & k = n-1 \end{cases}$$

Define $X_k \begin{cases} 1 & \text{if RandPartition gives } k \text{ vs } n-k-1 \text{ split.} \\ 0 & \text{otherwise} \end{cases}$

always assuming worst scenario

$$T(n) \leq \Theta(n) + \text{one of : } \begin{cases} T(\max\{0, n-1\}) \\ T(\max\{1, n-2\}) \\ T(\max\{2, n-3\}) \\ \vdots \\ T(\max\{n-1, 0\}) \end{cases}$$

$$\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})$$

$$E\left[T(n)\right] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T\left(\max\{k, n-k-1\}\right)\right]$$

$$E\left[T(n)\right] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E\left[\Theta(n)\right] + \sum_{k=0}^{n-1} E\left[X_k \cdot T(\max\{k, n-k-1\})\right] \qquad \text{why?}$$

$$E\left[T(n)\right] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E\left[\Theta(n)\right] + \sum_{k=0}^{n-1} E\left[X_k \cdot T(\max\{k, n-k-1\})\right]$$ by linearity of expectation

$$E[T(n)] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E[\Theta(n)] + \sum_{k=0}^{n-1} E[X_k \cdot T(\max\{k, n-k-1\})]$$ by linearity of expectation

$$= \Theta(n) + \sum \underbrace{E[X_k]} \cdot \underbrace{E[T(\max\{k, n-k-1\})]}$$ why?

$$E\left[T(n)\right] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E\left[\Theta(n)\right] + \sum_{k=0}^{n-1} E\left[X_k \cdot T(\max\{k, n-k-1\})\right]$$ by linearity of expectation

$$= \Theta(n) + \sum E[X_k] \cdot E\left[T(\max\{k, n-k-1\})\right]$$ by independence

$$E[T(n)] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E[\Theta(n)] + \sum_{k=0}^{n-1} E[X_k \cdot T(\max\{k, n-k-1\})]$$ by linearity of expectation

$$= \Theta(n) + \sum \underbrace{E[X_k]} \cdot E[T(\max\{k, n-k-1\})]$$ by independence

$$= \Theta(n) + \sum_{k=0}^{n-1} \cdot \underbrace{\frac{1}{n}} \cdot E[T(\max\{k, n-k-1\})]$$ by random choice

$$E\left[T(n)\right] \leq E\left[\Theta(n) + \sum_{k=0}^{n-1} X_k \cdot T(\max\{k, n-k-1\})\right]$$

$$= E\left[\Theta(n)\right] + \sum_{k=0}^{n-1} E\left[X_k \cdot T(\max\{k, n-k-1\})\right]$$ by linearity of expectation

$$= \Theta(n) + \sum E[X_k] \cdot E\left[T(\max\{k, n-k-1\})\right]$$ by independence

$$= \Theta(n) + \sum_{k=0}^{n-1} \cdot \frac{1}{n} \cdot E\left[T(\max\{k, n-k-1\})\right]$$ by random choice

$$\leq \Theta(n) + \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E\left[T(k)\right]$$

$$E\left[T(n)\right] \leqslant \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E\left[T(k)\right]$$

$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

↳ assume true for $k < n$

$$E[T(n)] \leq \Theta(n) + \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

assume true for $k<n$

$$\leq \Theta(n) + \frac{2}{n} \sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck$$

$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

assume true for $k<n$

$$\leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} ck \quad = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} k$$

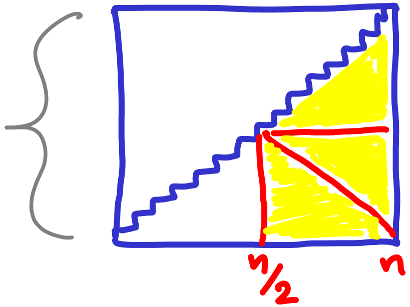
$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

assume true for $k < n$

$$\leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k$$

$$\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k \leq \frac{3}{8}n^2$$



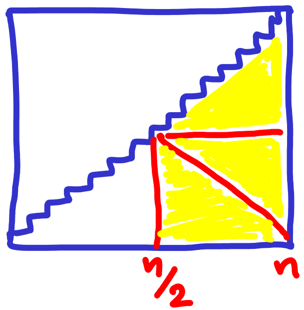
$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

assume true for $k<n$

$$\leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k$$

$$\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k \leq \frac{3}{8}n^2$$



$$\leq \Theta(n) + \frac{2c}{n} \cdot \frac{3}{8}n^2 = \Theta(n) + \frac{3cn}{4}$$

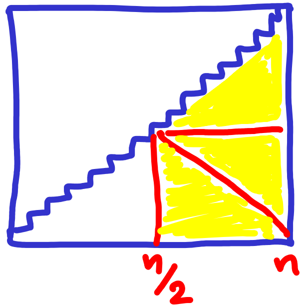
$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

assume true for $k<n$

$$\leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} ck = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} k$$

$$\sum_{k=\lfloor\frac{n}{2}\rfloor}^{n-1} k \leq \frac{3}{8}n^2$$



$$\leq \Theta(n) + \frac{2c}{n}\cdot\frac{3}{8}n^2 = \Theta(n) + \frac{3cn}{4} = cn - \frac{1}{4}cn + dn$$

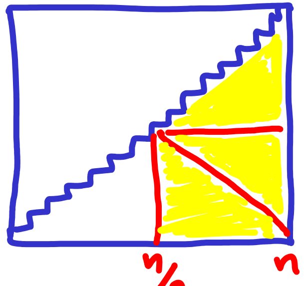
$$E[T(n)] \le \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \le c \cdot n$

assume true for $k<n$

$$\le \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k$$

$$\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k \le \frac{3}{8}n^2$$



$$\le \Theta(n) + \frac{2c}{n}\cdot\frac{3}{8}n^2 = \underline{\Theta(n)} + \frac{3cn}{4} = cn - \frac{1}{4}cn + dn$$

$$= cn - \underbrace{\left(\frac{c}{4} - d\right)}_{\text{positive if } c>4d} n$$

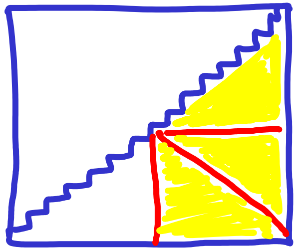
$$E[T(n)] \leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} E[T(k)]$$

Guess $E[T(n)] \leq c \cdot n$

↳ assume true for $k<n$

$$\leq \Theta(n) + \frac{2}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} ck = \Theta(n) + \frac{2c}{n}\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k$$

$$\sum_{k=\lfloor \frac{n}{2} \rfloor}^{n-1} k \leq \frac{3}{8}n^2$$



$$\leq \Theta(n) + \frac{2c}{n}\cdot\frac{3}{8}n^2 = \underline{\Theta(n)} + \frac{3cn}{4} = cn - \frac{1}{4}cn + dn$$

$$= cn - \underbrace{\left(\frac{c}{4} - d\right)}_{\text{positive if } c>4d} n$$

$$\boxed{E[T(n)] \leq 4dn}$$