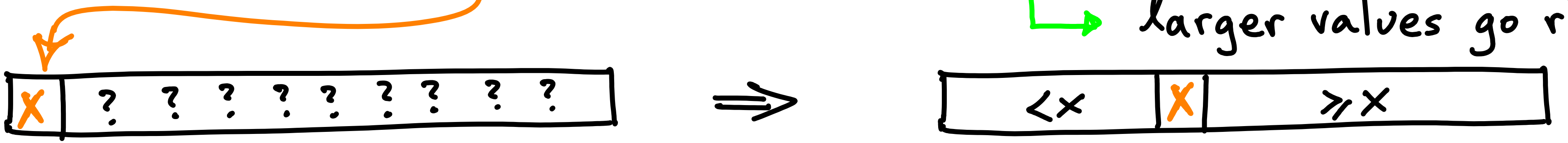
QUICKSORT

An in-place divide & conquer algorithm

QUICKSORT

An in-place divide & conquer algorithm

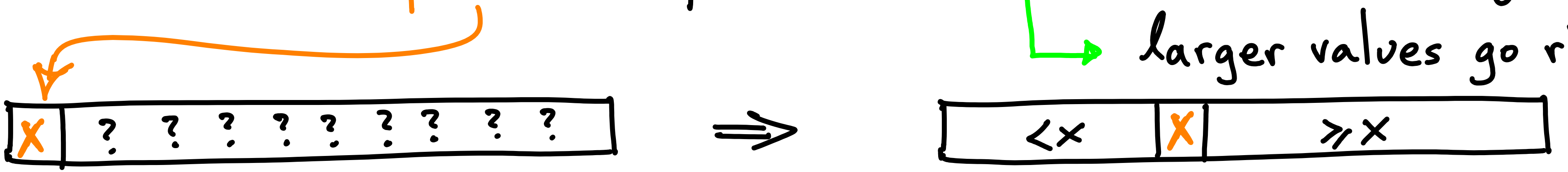
- DIVIDE: choose a pivot & partition → smaller values go left
  → larger values go right



QUICKSORT

An in-place divide & conquer algorithm

- DIVIDE: choose a pivot & partition
  - smaller values go left
  - larger values go right

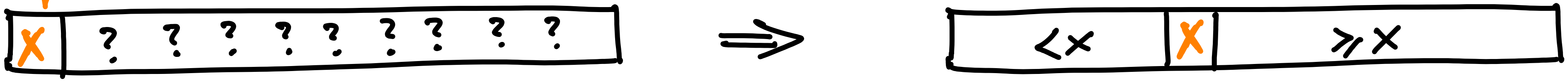


- CONQUER: Quicksort each side

QUICKSORT

An in-place divide & conquer algorithm

- DIVIDE: choose a pivot & partition
  - smaller values go left
  - larger values go right



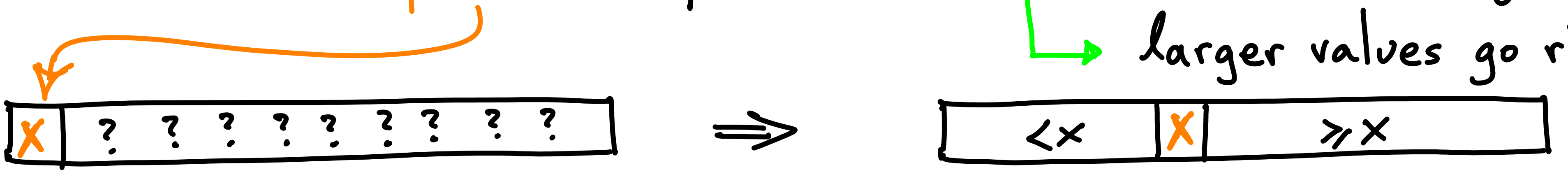
- CONQUER: Quicksort each side

Unlike Mergesort, there is no Combine phase.

QUICKSORT

An in-place divide & conquer algorithm

- DIVIDE: choose a pivot & partition → smaller values go left
  → larger values go right



- CONQUER: Quicksort each side

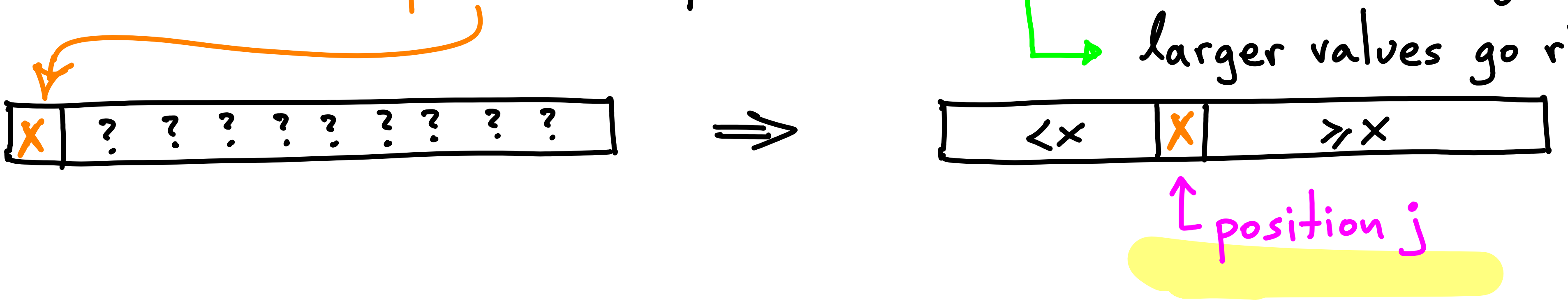
Unlike Mergesort, there is no Combine phase.

$T(n) = ?$

QUICKSORT

An in-place divide & conquer algorithm

- DIVIDE: choose a pivot & partition → smaller values go left
  → larger values go right



- CONQUER: Quicksort each side

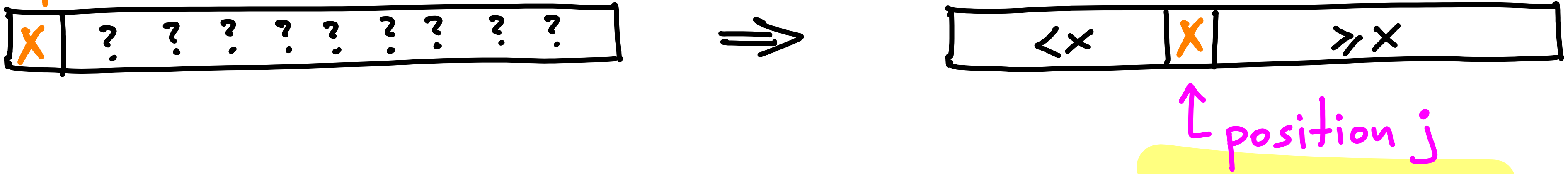
Unlike Mergesort, there is no Combine phase.

$$T(n) = ?$$

QUICKSORT

An in-place divide & conquer algorithm

- DIVIDE: choose a pivot & partition → smaller values go left
  → larger values go right



- CONQUER: Quicksort each side

Unlike Mergesort, there is no Combine phase.

$$T(n) = \Theta(n) + T(j-1) + T(n-j)$$

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	15	20	30	5	12	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	15	20	30	5	12	29	14

... continue

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	15	20	30	5	12	29	14

... continue

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	15	20	30	5	12	29	14

... continue

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	5	20	30	15	12	29	14

... continue

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	5	20	30	15	12	29	14

... continue

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	5	20	30	15	12	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

10	8	3	2	7	5	20	30	15	12	29	14

- DIVIDE: choose a pivot & partition

arbitrarily choose first

10	8	3	12	7	15	20	30	5	2	29	14

Grow "prefix" of smaller elements

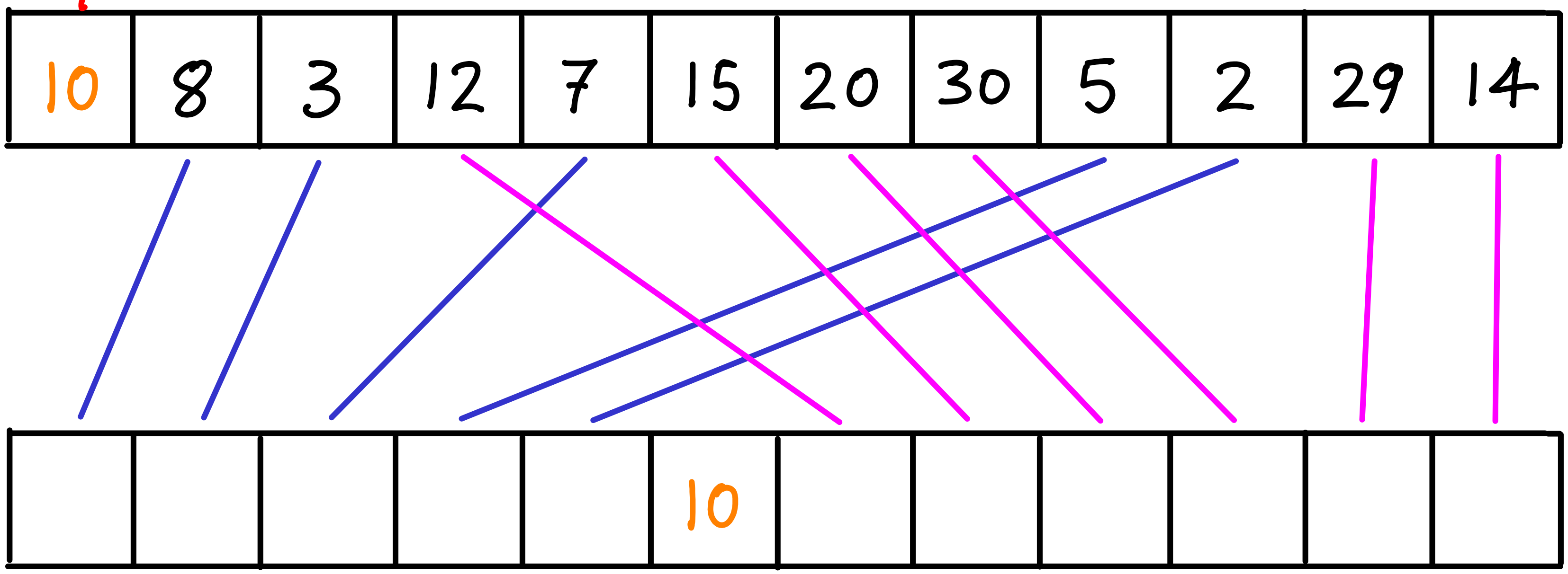
Grow suffix of larger elements

10	8	3	12	7	15	20	30	5	2	29	14

Now either the two sides meet or we can SWAP

5	8	3	2	7	10	20	30	15	12	29	14

- DIVIDE: choose a pivot & partition



There are stable versions as well

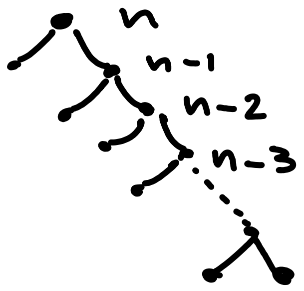
What is the worst-case time complexity, and why?

$T(n) = \Theta(n) + T(j-1) + T(n-j)$

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

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

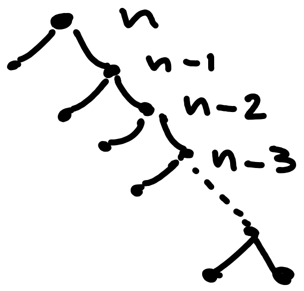
$$T(n) = T(0) + T(n-1) + \Theta(n) \quad = \Theta(n^2)$$



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

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

$$T(n) = T(0) + T(n-1) + \Theta(n) \quad = \Theta(n^2)$$

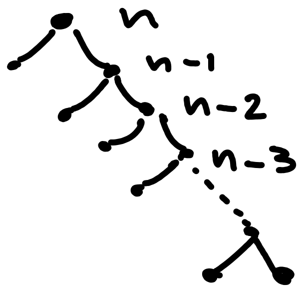


What would be ideal ?

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

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

$$T(n) = T(0) + T(n-1) + \Theta(n) \quad = \Theta(n^2)$$



What would be ideal ?

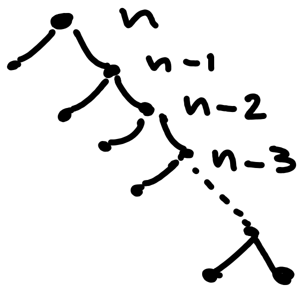
↳ ~ even split, every time

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

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

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

$$T(n) = T(0) + T(n-1) + \Theta(n) = \Theta(n^2)$$



What would be ideal ?

↳ ~ even split, every time

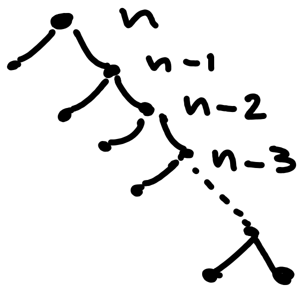
$$T(n) \leq 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \log n)$$

Why use Quicksort?

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

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

$$T(n) = T(0) + T(n-1) + \Theta(n) = \Theta(n^2)$$



What would be ideal ?

↳ ~ even split, every time

$$T(n) \leq 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \log n)$$

Why use Quicksort? We expect $\Theta(n \log n)$ ... with a small constant

(& it's in-place, and stable)

What if we always split "sort-of-evenly"?

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

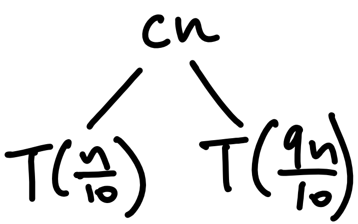
What if we always split "sort-of-evenly"?

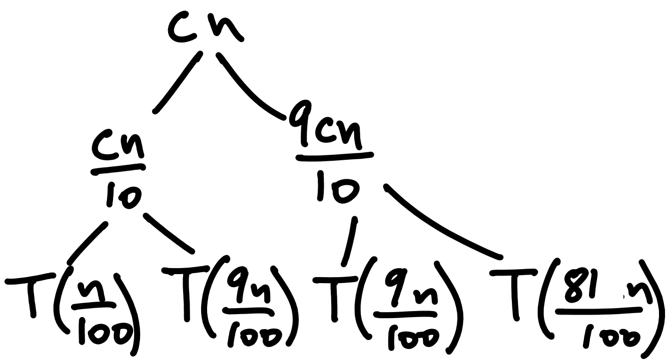
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$

$$\begin{array}{ccc} & cn & \\ \swarrow & & \searrow \\ T\left(\frac{n}{10}\right) & & T\left(\frac{9n}{10}\right) \end{array}$$

What if we always split "sort-of-evenly"?

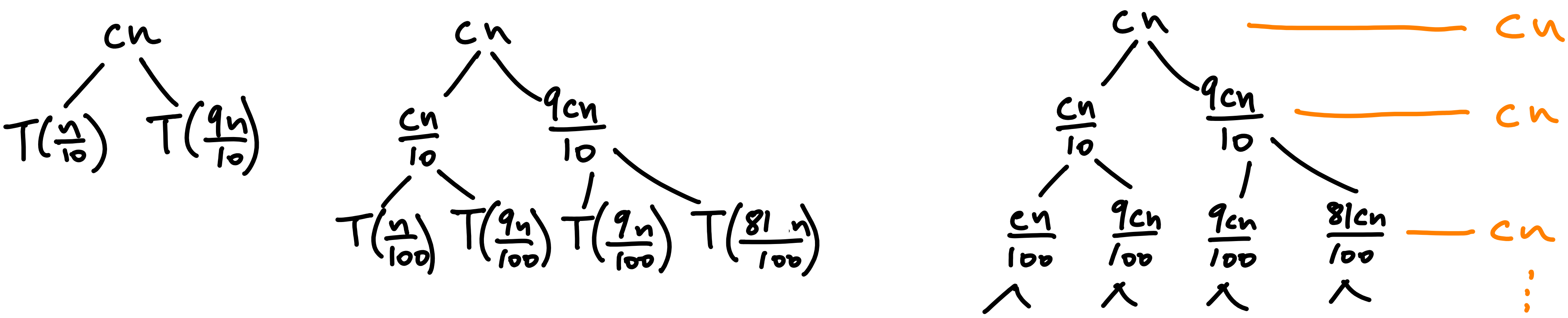
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$





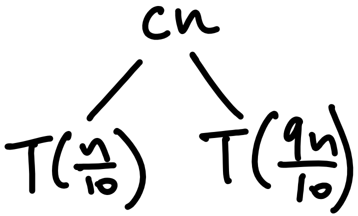
What if we always split "sort-of-evenly"?

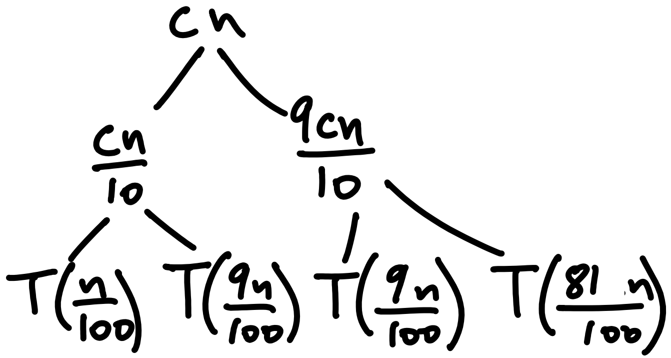
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$

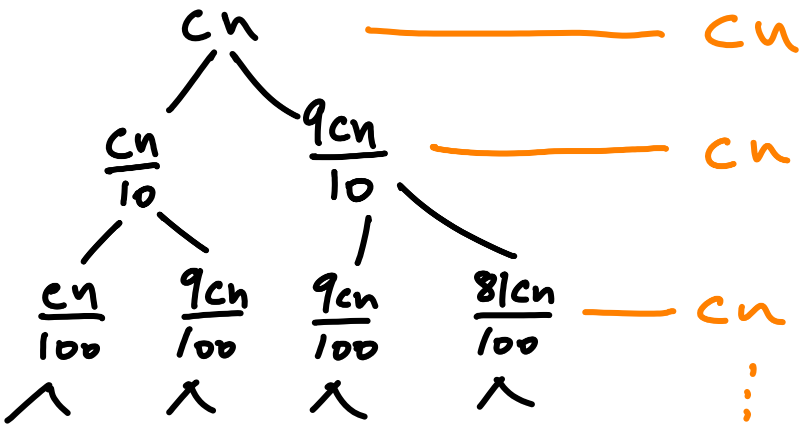


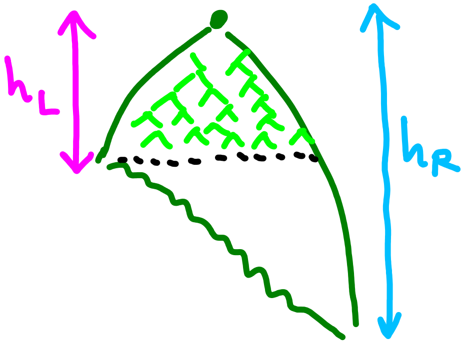
What if we always split "sort-of-evenly"?

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



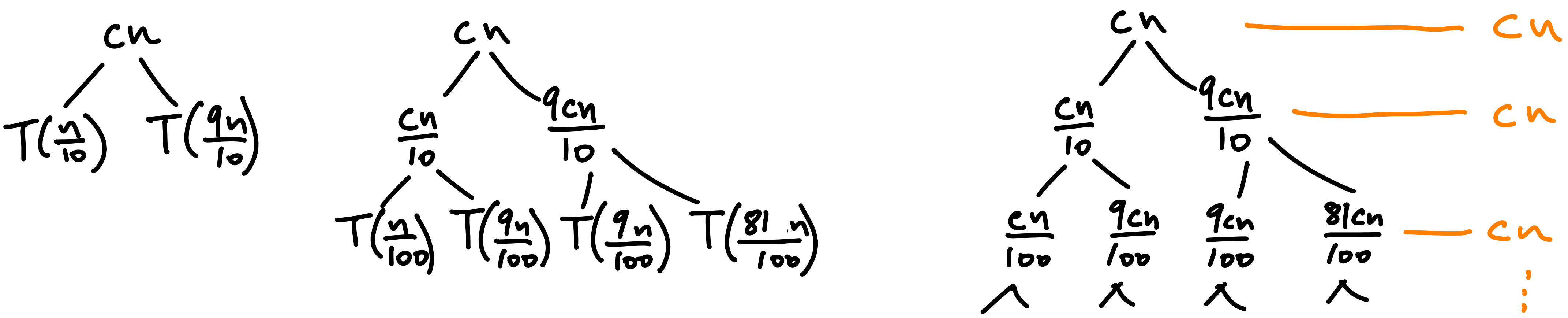


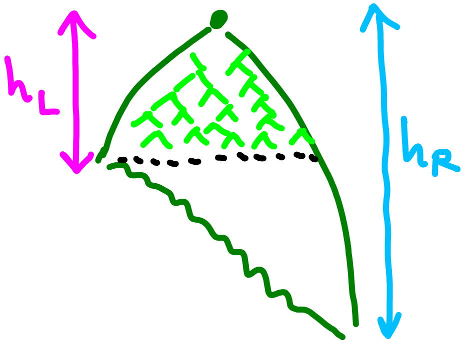




What if we always split "sort-of-evenly"?

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



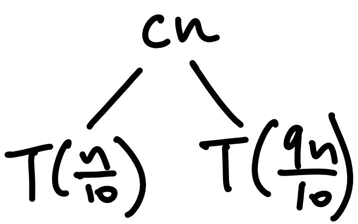


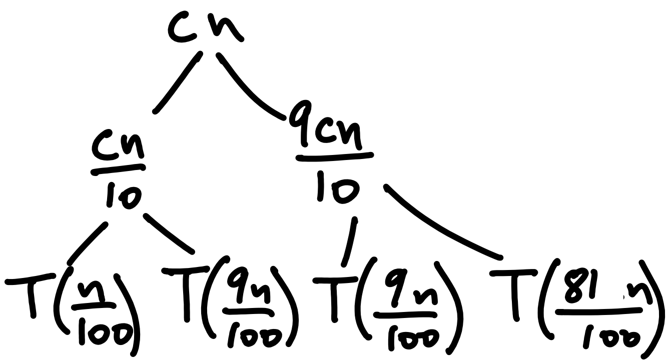
$h_L \sim \log_{10} n$

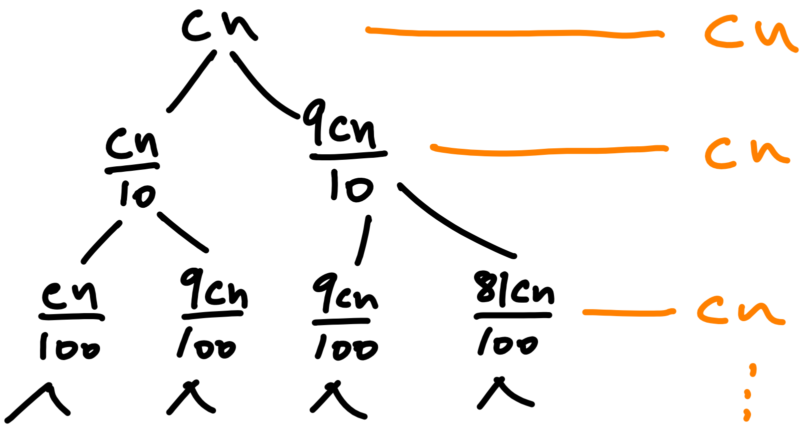
$h_R \sim \log_{10/9} n$

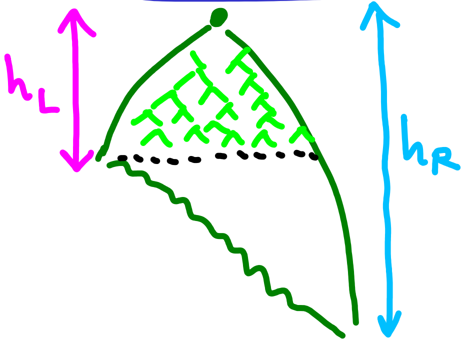
What if we always split "sort-of-evenly"?

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







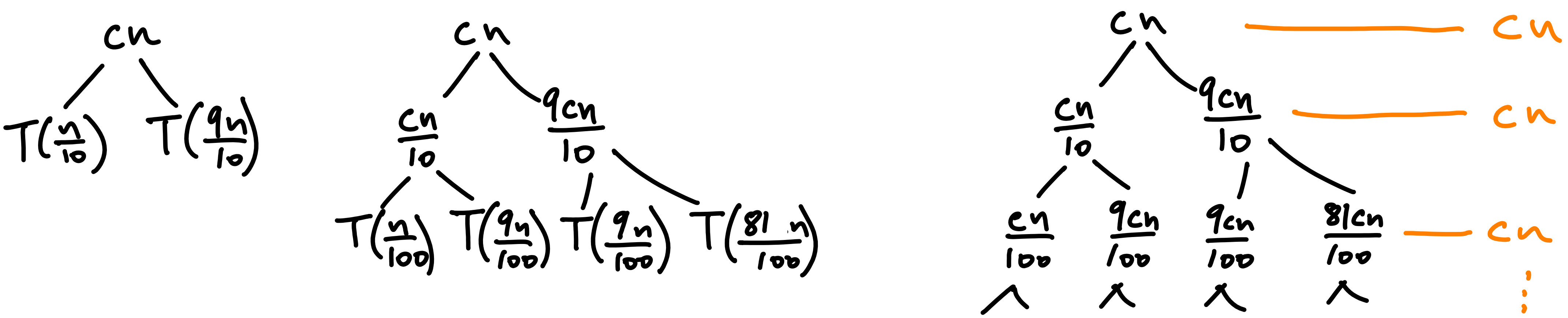


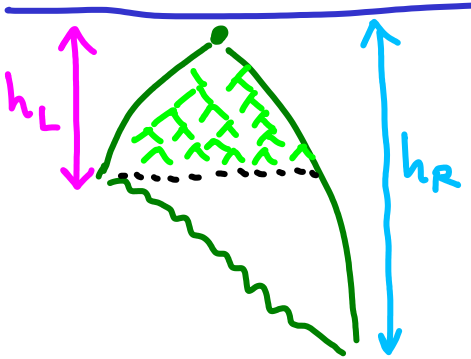
$h_L \sim \log_{10} n \rightarrow T(n) \geq cn \log_{10} n$

$h_R \sim \log_{10/9} n \rightarrow T(n) \leq cn \log_{10/9} n$

What if we always split "sort-of-evenly"?

e.g., $T(n) = T(\frac{n}{10}) + T(\frac{9n}{10}) + c \cdot n$





$h_L \sim \log_{10} n \rightarrow T(n) \geq cn \log_{10} n$

$h_R \sim \log_{10/9} n \rightarrow T(n) \leq cn \log_{10/9} n$

Any constant-fraction-split will give $\Theta(n \log n)$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$L(n) = 2U\left(\frac{n}{2}\right) + dn$$

$$U(n) = L(n-1) + dn$$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\frac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2 \cdot \left[\ ?\ ?\ ?\ \right] + dn$$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\,\tfrac{n}{2}\right] + dn$$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \quad = \Theta(n\log n)$$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \qquad = \Theta(n\log n)$$

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

unbalanced otherwise

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\,\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \quad = \Theta(n\log n)$$

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: $T(n) \leq$?

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \quad = \Theta(n\log n)$$

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

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

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \quad = \Theta(n\log n)$$

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

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

Worst case if unbalanced split: $T(n) \leq T(n-1) + T(0) + dn$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \qquad = \Theta(n \log n)$$

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

Worst case if unbalanced split: $T(n) \leq T(n-1) + T(0) + dn \leq T(n) + dn$

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{array}{l} L(n) = 2U\left(\frac{n}{2}\right) + dn \\ U(n) = L(n-1) + dn \end{array}\right\} \quad L(n) = 2\cdot\left[L\left(\frac{n}{2}-1\right) + d\frac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\frac{n}{2}\right) + 2dn \quad = \Theta(n \log n)$$

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

Worst case if unbalanced split: $T(n) \leq T(n-1) + T(0) + dn \leq T(n) + dn$

Each split has a 50% chance of being balanced

What if we alternate between ideal & bad splits? (Lucky vs Unlucky)

$$\left.\begin{aligned} L(n) &= 2U\left(\tfrac{n}{2}\right) + dn \\ U(n) &= L(n-1) + dn \end{aligned}\right\} \quad L(n) = 2\cdot\left[L\left(\tfrac{n}{2}-1\right) + d\tfrac{n}{2}\right] + dn$$

$$\leq 2\cdot L\left(\tfrac{n}{2}\right) + 2dn \qquad = \Theta(n \log n)$$

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

Worst case if unbalanced split: $T(n) \leq T(n-1) + T(0) + dn \leq \underline{T(n) + dn}$

Each split has a 50% chance of being balanced

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

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

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

Each split has a 50% chance of being balanced

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

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

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

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

Each split has a 50% chance of being balanced

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

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

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

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

Each split has a 50% chance of being balanced

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

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

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