QUICKSORT

QUICKSORT

- If necessary shuffle data to make random order.

QUICKSORT

- If necessary shuffle data to make random order.
- DIVIDE: choose a pivot & partition.

QUICKSORT

- If necessary shuffle data to make random order.

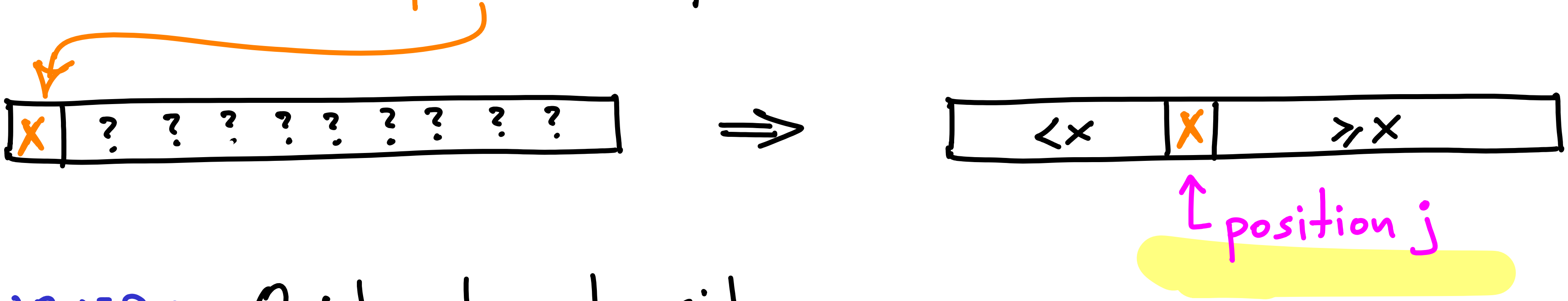
- DIVIDE: choose a pivot & partition.

- CONQUER: Quicksort each side.

QUICKSORT

- If necessary shuffle data to make random order.

- DIVIDE: choose a pivot & partition.



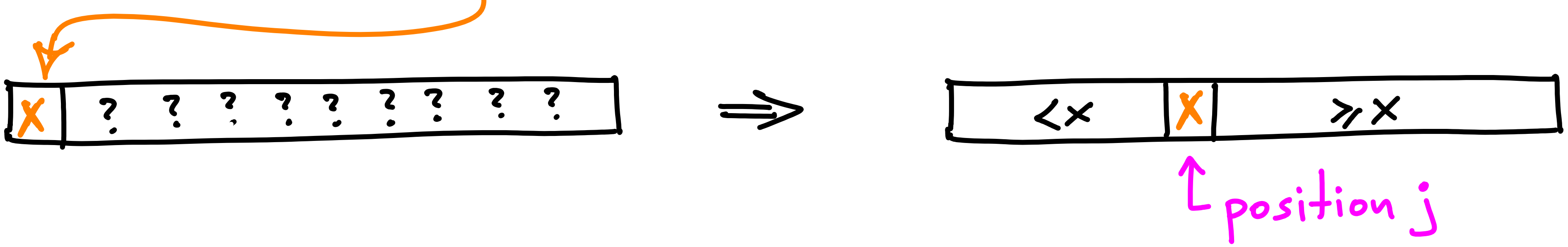
- CONQUER: Quicksort each side.

$$T(n) = \quad ?$$

QUICKSORT

- If necessary shuffle data to make random order.

- DIVIDE: choose a pivot & partition. $\rightarrow \Theta(n)$



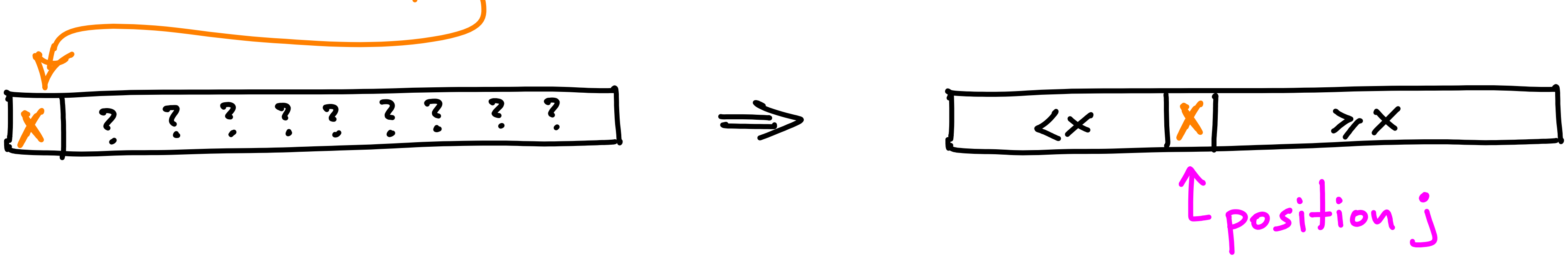
- CONQUER: Quicksort each side.

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

QUICKSORT

- If necessary shuffle data to make random order.

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



- **CONQUER:** Quicksort each side.

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

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

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...

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

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

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

$T(n) =$?

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

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

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

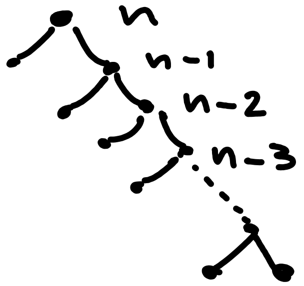
$$T(n) = T(0) + T(n-1) + \Theta(n) \quad = \quad ?$$

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

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

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

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

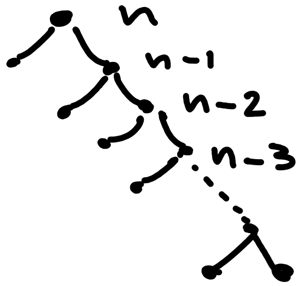


$$T(n) = \Theta(n) + T(j-1) + T(n-j)$$

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

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

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



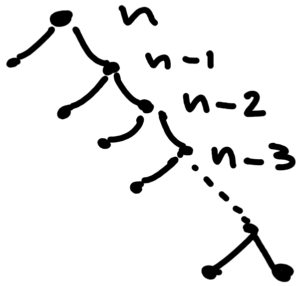
What would be ideal ?

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

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

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

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



What would be ideal ?

$\hookrightarrow$ ~ balanced partition, every time

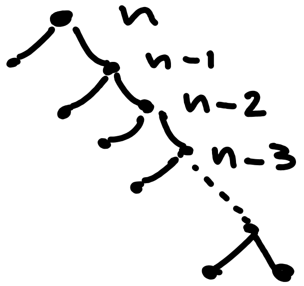
$$T(n) = ?$$

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

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 would be ideal ?

↳ ~ balanced partition, every time

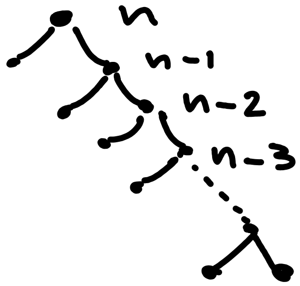
$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n) \quad = ?$$

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

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

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

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



What would be ideal?

↳ ~ balanced partition, every time

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

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

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

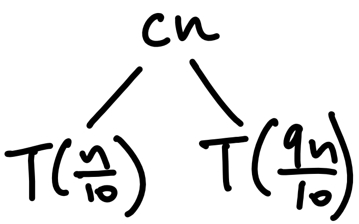
What if we always have a "sort-of-balanced" partition?

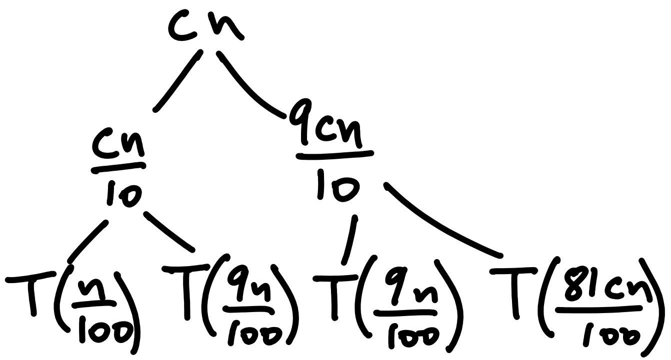
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$

$$
\begin{tikzcd}[column sep=small]
 & cn \arrow[dl, dash] \arrow[dr, dash] & \\
T\left(\frac{n}{10}\right) & & T\left(\frac{9n}{10}\right)
\end{tikzcd}
$$

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

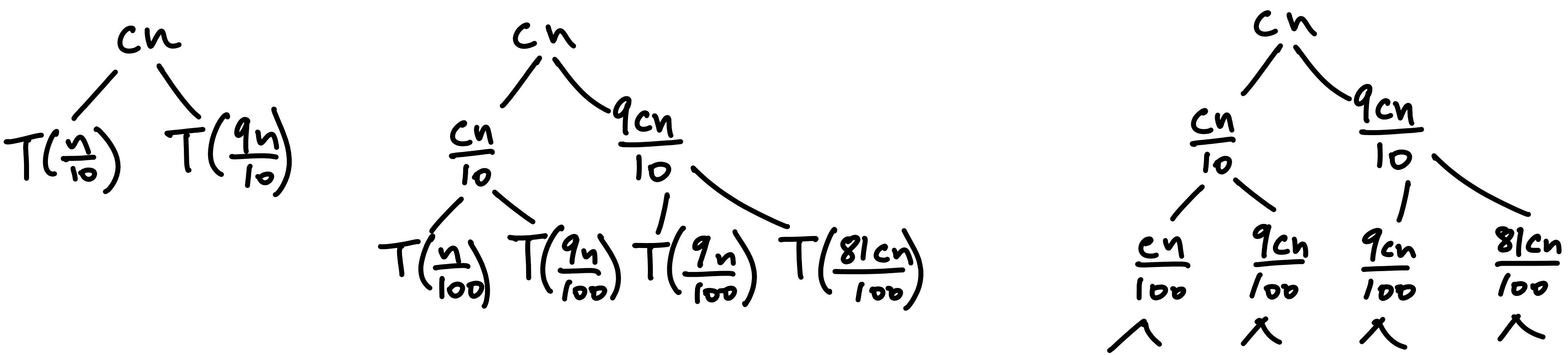
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$





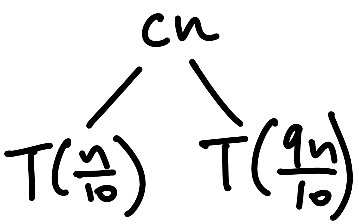
What if we always have a "sort-of-balanced" partition?

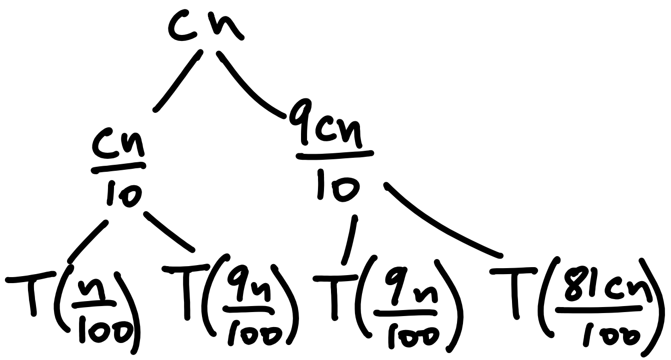
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$

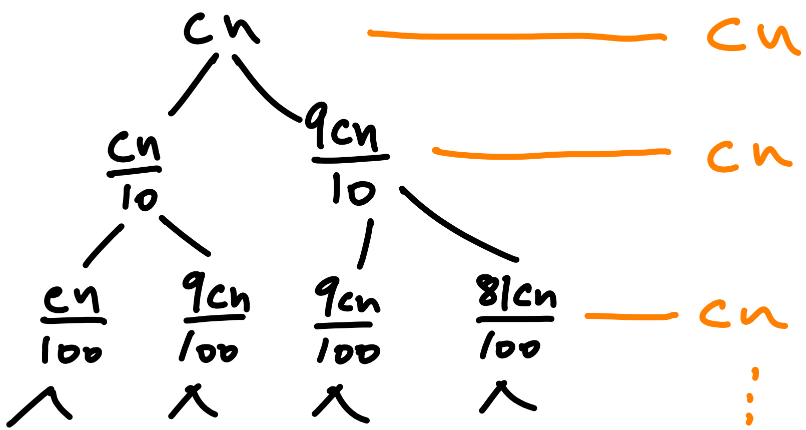


What if we always have a "sort-of-balanced" partition?

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

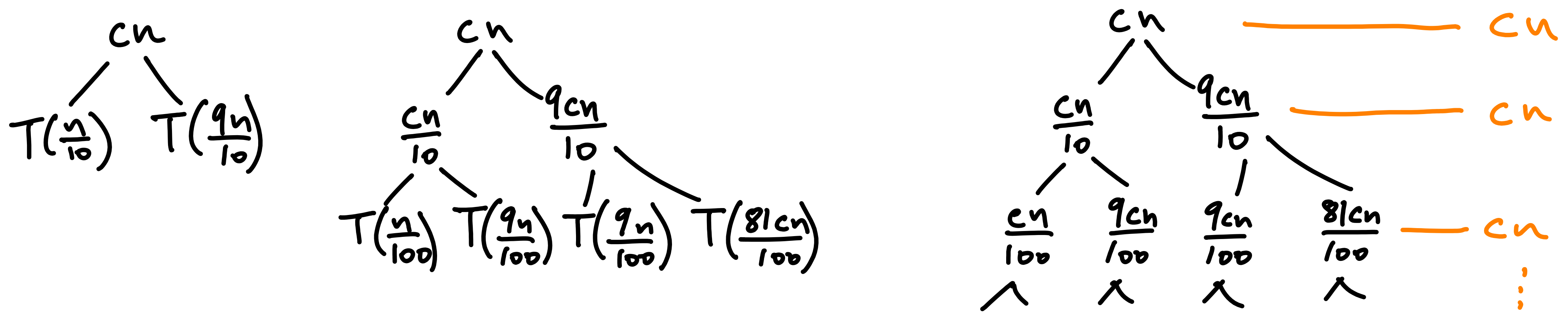


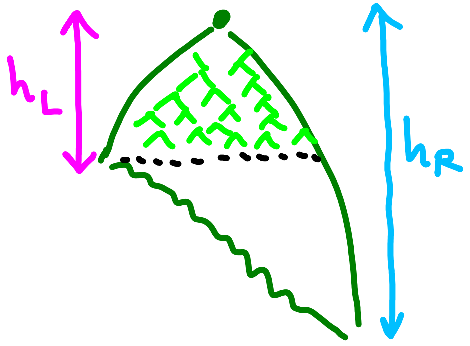




What if we always have a "sort-of-balanced" partition?

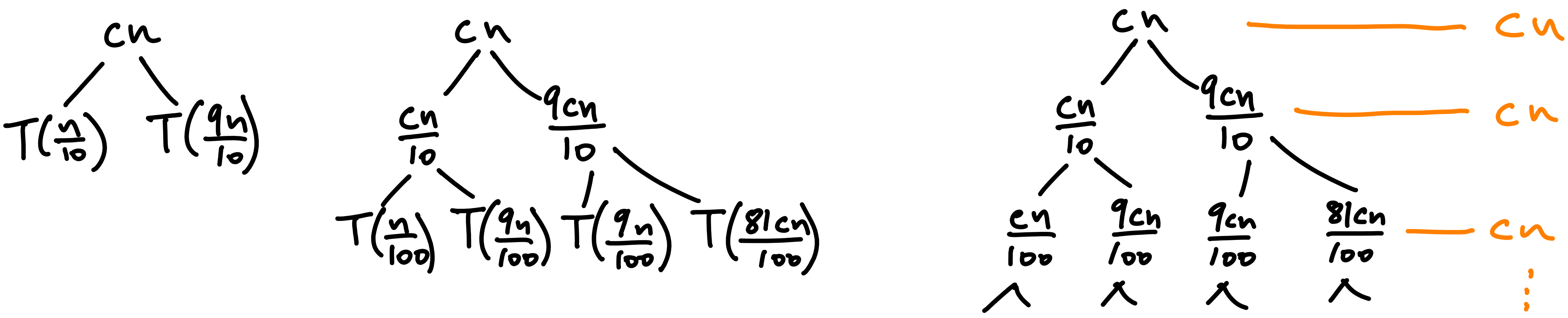
e.g., $T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + c \cdot n$

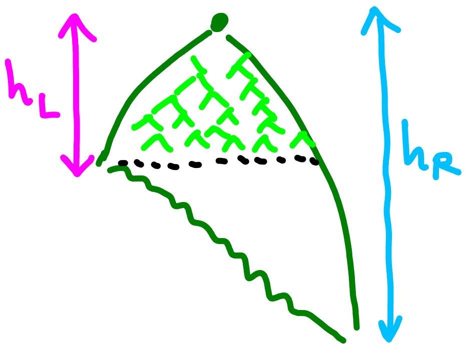




What if we always have a "sort-of-balanced" partition?

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



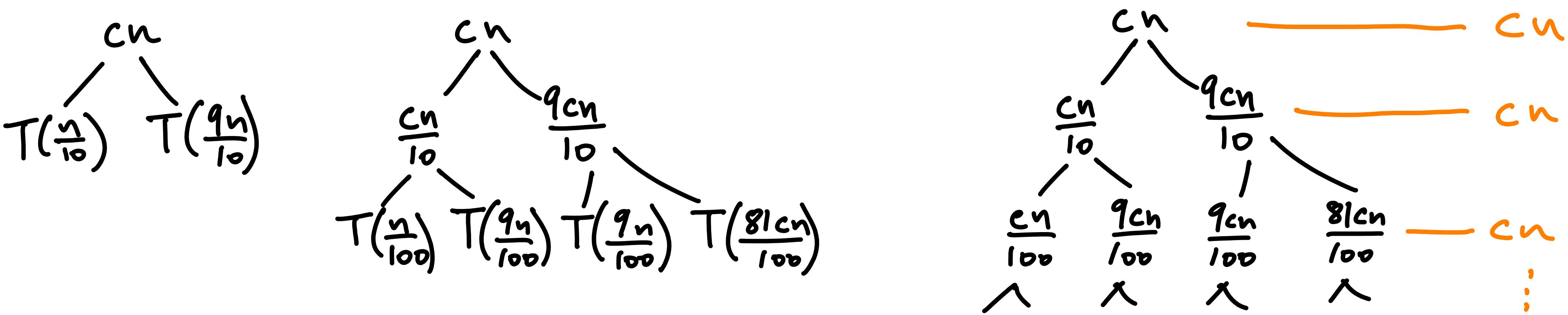


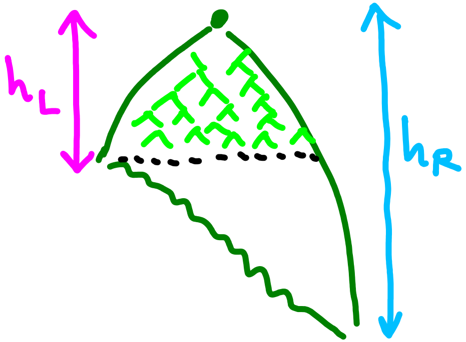
$h_L \sim \log_{10} n$

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

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

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



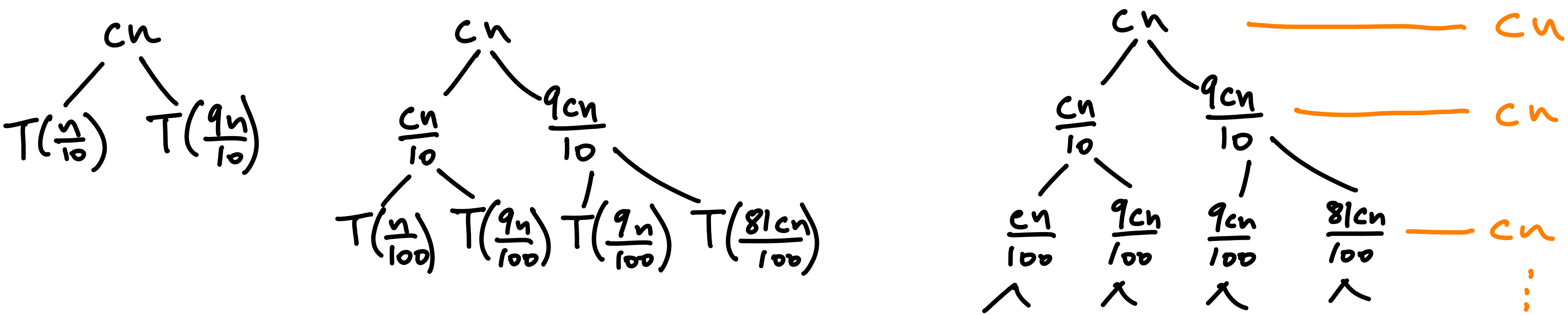


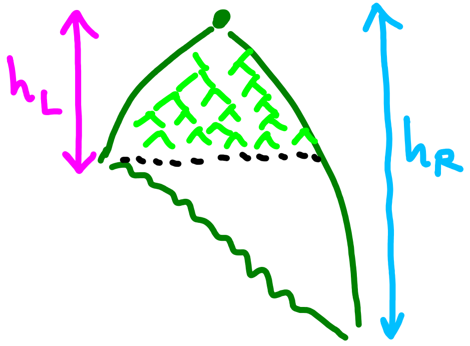
$h_L \sim \log_{10} n \rightarrow T(n) \geq cn \log_{10} n$

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

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

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



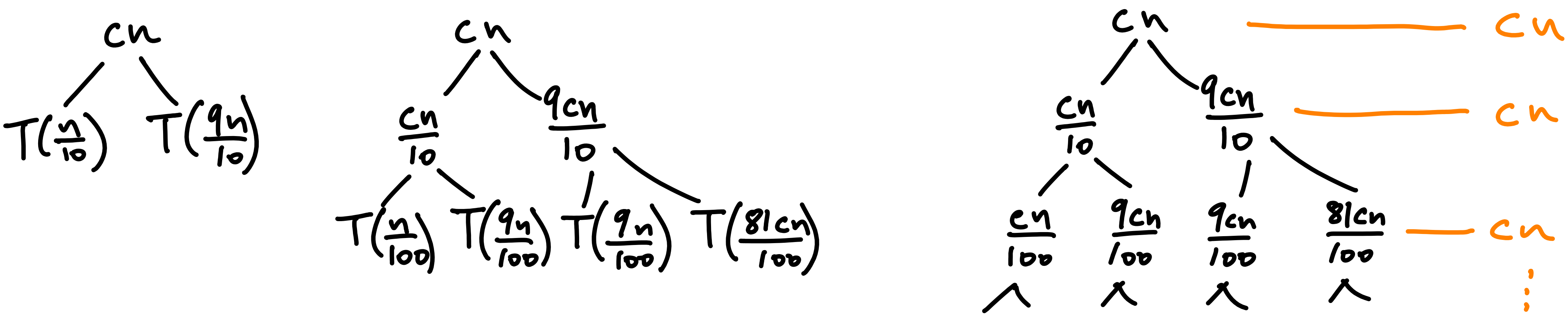


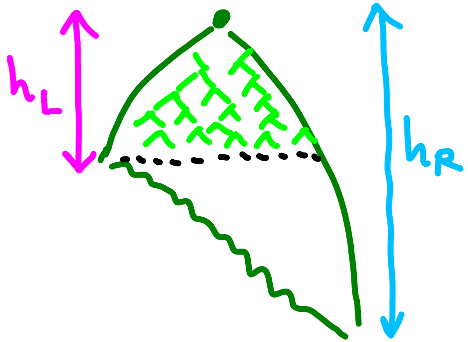
$h_L \sim \log_{10} n \rightarrow T(n) \geq cn \log_{10} n$

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

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

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





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

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

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

Expected time

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

(redefining what balanced means)

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

→ unbalanced otherwise

(redefining what balanced means)

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

$E[T(n)]$

(redefining what balanced means)

↳ but let's just use $T(n)$

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

→ unbalanced otherwise

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

$\rightarrow$ unbalanced otherwise

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

→ unbalanced otherwise

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

Expected time: $E[T(n)]$

but let's just use $T(n)$

call a partition $\rightarrow$ balanced if pivot ranks in $[\frac{n}{4} \ldots \frac{3n}{4}]$

$\rightarrow$ unbalanced otherwise

(redefining what balanced means)

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

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

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

$E[T(n)]$ → unbalanced otherwise

but let's just use $T(n)$ (redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

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

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

$T(n) \leq ?$

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

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

$$T(n) \leq \frac{1}{2}\left(T(n)+cn\right) + \frac{1}{2}\left(T(\frac{3n}{4}) + T(\frac{n}{4}) + cn\right)$$

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

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

$$T(n) \leq \frac{1}{2}\left(T(n)+cn\right) + \frac{1}{2}\left(T(\tfrac{3n}{4}) + T(\tfrac{n}{4}) + cn\right)$$

$$\frac{1}{2}T(n) \leq cn + \frac{1}{2}\left(T(\tfrac{3n}{4}) + T(\tfrac{n}{4})\right)$$

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

→ unbalanced otherwise

$E[T(n)]$

but let's just use $T(n)$

(redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

$$T(n) \leq \frac{1}{2}\left(T(n)+cn\right) + \frac{1}{2}\left(T(\frac{3n}{4}) + T(\frac{n}{4}) + cn\right)$$

$$\frac{1}{2}T(n) \leq cn + \frac{1}{2}\left(T(\frac{3n}{4}) + T(\frac{n}{4})\right)$$

$$T(n) \leq T(\frac{3n}{4}) + T(\frac{n}{4}) + 2cn$$

Expected time: $E[T(n)]$ but let's just use $T(n)$

call a partition $\rightarrow$ balanced if pivot ranks in $[\frac{n}{4} \ldots \frac{3n}{4}]$
$\rightarrow$ unbalanced otherwise

(redefining what balanced means)

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

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

Each partition has a 50% chance of being balanced

$$T(n) \leq \frac{1}{2}(T(n)+cn) + \frac{1}{2}\left(T(\tfrac{3n}{4}) + T(\tfrac{n}{4}) + cn\right)$$

$$\frac{1}{2}T(n) \leq cn + \frac{1}{2}\left(T(\tfrac{3n}{4}) + T(\tfrac{n}{4})\right)$$

$$T(n) \leq T(\tfrac{3n}{4}) + T(\tfrac{n}{4}) + 2cn \quad = O(n\log n)$$