QUICKSORT

An in-place divide & conquer algorithm

-DIVIDE: choose a pivot & partition

\[ \begin{array}{cccccccccc}
\end{array} \]

\[ \Rightarrow \]

\[ \begin{array}{cccccccccc}
<x & X & >x
\end{array} \]

-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

Grow "prefix" of smaller elements

Grow suffix of larger elements

Now either the two sides meet or we can swap

... continue
- **DIVIDE**: choose a pivot & partition

  arbitrarily choose first

  Grow “prefix” of smaller elements  
  Grow suffix of larger elements

  Now either the two sides meet or we can **SWAP**

  ...continue
- **DIVIDE:** choose a pivot & partition

  arbitrarily choose first

  Grow "prefix" of smaller elements

  Grow suffix of larger elements

  Now either the two sides meet or we can **SWAP**

  ... continue
Divide: choose a pivot & partition arbitrarily choose first

Grow "prefix" of smaller elements  Grow suffix of larger elements

Now either the two sides meet or we can swap

...continue
- **DIVIDE**: choose a pivot & partition
  arbitrarly choose first

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

Grow "prefix" of smaller elements
Grow suffix of larger elements

Now either the two sides meet or we can SWAP
- **DIVIDE:** choose a **pivot** & partition

There are stable versions as well
What is the worst-case time complexity, and why?

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

What would be ideal?

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

Why use Quicksort? We expect \( \Theta(n \log n) \) ... with a small constant (and it's in-place, and stable)
What if we **always** split "sort-of-evenly"?

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

Any constant-fraction-split will give \( \Theta(n\log n) \)
What if we alternate between **ideal** & **bad** splits?  \( \text{ (Lucky vs Unlucky) } \)

\[
\begin{align*}
L(n) &= 2U\left(\frac{n}{2}\right) + dn \\
U(n) &= L(n-1) + dn
\end{align*}
\]

\[
L(n) = 2 \left[ L\left(\frac{n}{2} - 1\right) + d \frac{n}{2} \right] + dn \\
\leq 2 \cdot L\left(\frac{n}{2}\right) + 2dn = \Theta(n \log n)
\]
Expected time: call a split balanced if pivot ranks in $[\frac{n}{4} \ldots \frac{3n}{4}]$
unbalanced otherwise

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

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

Each split has a 50% chance of being balanced

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

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

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