Warm up

Compare $f(n + m)$ with $f(n) + f(m)$

When $f(n) = O(n)$

When $f(n) = \Omega(n)$
$f(n) \in O(n)$

$f(n + m) \leq f(n) + f(m)$
\[ f(n) \in \Omega(n) \]

\[ f(n + m) \geq f(n) + f(m) \]
\[ f(n) = \Theta(n) \]

\[ f(n + m) = f(n) + f(m) \]
Partition (Divide step)

Given: a list, a pivot $p$
Start: unordered list

Goal: All elements $< p$ on left, all $> p$ on right
Conquer

Recursively sort Left and Right sublists

All elements < $p$

All elements > $p$

Exactly where it belongs!
Quicksort Run Time (Best)

If the **pivot** is always the median:

Then we divide in half each time

\[ T(n) = 2T\left(\frac{n}{2}\right) + n \]

\[ T(n) = O(n \log n) \]
Quicksort on a (nearly) Sorted List

First element always yields unbalanced pivot

So we shorten by 1 each time

$$T(n) = T(n - 1) + n$$

$$T(n) = O(n^2)$$
Quickselect

• Finds $i^{th}$ order statistic
  – $i^{th}$ smallest element in the list
  – $1^{st}$ order statistic: minimum
  – $n^{th}$ order statistic: maximum
  – $\frac{n}{2}^{th}$ order statistic: median
Partition (Divide step)

Given: a list, a pivot value \( p \)

Start: unordered list

Goal: All elements \(< p\) on left, all \(> p\) on right
Conquer

Exactly where it belongs!

All elements < $p$

All elements > $p$

Recurse on sublist that contains index $i$
(adjust $i$ accordingly if recursing right)
Quickselect Run Time

If the pivot is always the median:

Then we divide in half each time

\[ S(n) = S \left( \frac{n}{2} \right) + n \]

\[ S(n) = \mathcal{O}(n) \]
Quickselect Run Time

If the partition is always unbalanced:

Then we shorten by 1 each time

\[ S(n) = S(n - 1) + n \]

\[ S(n) = O(n^2) \]
Good Pivot

• What makes a good Pivot?
  – Roughly even split between left and right
  – Ideally: median

• Two Options:
  – Worst case $\Theta(n)$ algorithm (Median of Medians)
    • This algorithm uses Quickselect as a subroutine
  – Randomized algorithm in $\Theta(n)$ expected
QuickSort Run Time

If the pivot is always $\frac{n}{10}$th order statistic:

$$T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + n$$
\[ T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + n \]
Quicksort Run Time

If the pivot is always $\frac{n}{10}$th order statistic:

$$T(n) = T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right) + n$$

$$T(n) = \Theta(n \log n)$$
Good Pivot

• Our “good pivot”
  – Both sides of Pivot >30%

Or

Select Pivot from this range

>30%

>30%
Median of Medians

• Fast way to select a “good” pivot
• Guarantees pivot is greater than 30% of elements and less than 30% of the elements
• Idea: break list into chunks, find the median of each chunk, use the median of those medians
Median of Medians

1. Break list into chunks of size 5

2. Find the median of each chunk

3. Return median of medians (using Quickselect)
Why is this good?

Each chunk sorted, chunks ordered by their medians

MedianofMedians $\geq$ all of these

$\frac{n}{5}$

MedianofMedians $\leq$ all of these
Why is this good?

Larger than 3 things in each (but one) list to the left

\[ 3 \left( \frac{1}{2} \cdot \left\lfloor \frac{n}{5} \right\rfloor - 1 \right) \approx \frac{3n}{10} - 3 \text{ elements} \leq \]

Similarly:

\[ 3 \left( \frac{1}{2} \cdot \left\lfloor \frac{n}{5} \right\rfloor - 1 \right) \approx \frac{3n}{10} - 3 \text{ elements} \geq \]
Quickselect

- **Divide:** select an element \( p \) using Median of Medians, \( \text{Partition}(p) \)

- **Conquer:** if \( i = \text{index of } p \), done, if \( i < \text{index of } p \) recurse left. Else recurse right

- **Combine:** Nothing!

\[
M(n) + \Theta(n) \\
\leq S\left(\frac{7}{10}n\right) \\
S(n) \leq S\left(\frac{7}{10}n\right) + M(n) + \Theta(n)
\]
Median of Medians, Run Time

1. Break list into chunks of 5 \( \Theta(n) \)

2. Find the median of each chunk \( \Theta(n) \)

3. Return median of medians (using Quickselect)

\[
M(n) = S\left(\frac{n}{5}\right) + \Theta(n)
\]
Quickselect

\[ S(n) \leq S\left(\frac{7n}{10}\right) + M(n) + \Theta(n) \]

\[ M(n) = S\left(\frac{n}{5}\right) + \Theta(n) \]

\[ = S\left(\frac{7n}{10}\right) + S\left(\frac{n}{5}\right) + \Theta(n) \]

\[ = S\left(\frac{7n}{10}\right) + S\left(\frac{2n}{10}\right) + \Theta(n) \]

\[ \leq S\left(\frac{9n}{10}\right) + \Theta(n) \quad \text{Because } S(n) = \Omega(n) \]

Master theorem Case 3!

\[ S(n) = O(n) \]

\[ S(n) = \Theta(n) \]
Phew! Back to Quicksort

Using Quickselect, with a median-of-medians partition:

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

Then we divide in half each time

\[ T(n) = \Theta(n \log n) \]
Is it worth it?

• Using Quickselect to pick median guarantees $\Theta(n \log n)$ run time

• Approach has very large constants
  – If you really want $\Theta(n \log n)$, better off using MergeSort

• Better approach: Random pivot
  – Very small constant (very fast algorithm)
  – Expected to run in $\Theta(n \log n)$ time
    • Why? Unbalanced partitions are very unlikely
We only ever compare Begin and $p$

**Partition, Procedure**

If $\text{Begin value} < p$, move $\text{Begin}$ right
Else swap $\text{Begin}$ value with $\text{End}$ value, move $\text{End}$ Left
Done when $\text{Begin} = \text{End}$
Getting $n \log n$ Expected

• Remember, run time counts comparisons!
• Quicksort only compares against a **pivot**
  – Element $i$ only compared to element $j$ if one of them was the **pivot**
Formal Argument for $n \log n$ Average

• What is the probability of comparing two given elements?

• Let indicator random variable $C_{i,j} = 1$ if we compare index $i$ with index $j$.

• $\Pr[C_{3,4} = 1] = 1$
  - Why? Otherwise I wouldn’t know which came first
  - ANY sorting algorithm must compare adjacent elements
Formal Argument for $n \log n$ Average

- What is the probability of comparing two given elements?

- $\Pr[C_{1,12} = 1] = \frac{2}{12}$
  - Why?
    - I only compare 1 with 12 if either was chosen as the first pivot
    - Otherwise they would be divided into opposite sublists
Formal Argument for $n \log n$ Average

- $\Pr[C_{i,j} = 1]$ for $i < j$:
  - $j - i + 1$ elements in range $[i, j]$
  - Compared if $i$ or $j$ is first pivot chosen from $[i, j]$
  - $\Pr[C_{i,j} = 1] = \frac{2}{j-i+1}$

- Expected number of comparisons:
  - $\mathbb{E}[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{i,j}]$
  - $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j-i+1}$
Expected number of Comparisons

Consider when \( i = 1 \)

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}
\]

Compared if 1 or 2 are chosen as pivot
(these will always be compared)

Sum so far: \( \frac{2}{2} \)
Expected number of Comparisons

Consider when $i = 1$

Compared if 1 or 3 are chosen as pivot
(but never if 2 is ever chosen)

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}$$

$\begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\end{array}$

Sum so far: $\frac{2}{2} + \frac{2}{3}$
Expected number of Comparisons

Consider when $i = 1$

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}
\]

Compared if 1 or 4 are chosen as pivot (but never if 2 or 3 are chosen)

Sum so far: $\frac{2}{2} + \frac{2}{3} + \frac{2}{4}$
Expected number of Comparisons

Consider when $i = 1$

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}
\]

Compared if 1 or 12 are chosen as pivot
(but never if 2 -> 11 are chosen)

Overall sum: \(\frac{2}{2} + \frac{2}{3} + \frac{2}{4} + \frac{2}{5} + \cdots + \frac{2}{n}\)
Expected number of Comparisons

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1}
\]

When \( i = 1 \):

\[
2 \left( \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} \right)
\]

\( n \) terms overall

\[
\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{j - i + 1} \leq 2n \left( \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \right) \quad \Theta(\log n)
\]

Quicksort overall: expected \( \Theta(n \log n) \)