Median Finding

1. Testing iroot
2. Analyze backboneSimilar
3. Median finding
Testing iroot

• on interval 1, 2, 3, 4, 5 suppose function values for some procedure \( f \) are
• 7, -2, -8, 5, -3
• checkExpect(iroot(1, 4, f), ???) ?

```
let k = iroot(1, 4, f);
checkExpect(f(k)*f(k+1) <= 0, true);
```
Analyze backboneSimilar

- Let $B(n)$ be the max number of operations involved in evaluating backboneSimilar$(t_1, t_2)$, where $n$ is the number of nodes/leaves in the larger of $t_1$ and $t_2$.

\[ B(1) = a \]
\[ B(n) \leq c+ ? \]
Analyze backboneSimilar

\[ B(1) = a \]
\[ B(n) \leq c + ? \]

- We're going to apply backboneSimilar to the left and right subtrees. If the left subtree has \( k \) items, the right has \( n-k-1 \) (the -1 for the item at the current node!)
  - But we don't know what \( k \) is
  - Could be any number from 1 to \( n-1 \)
  - \( B(n) \leq c + \max_{k=1\ldots n-1} (B(k) + B(n - k - 1)) \)
Analyze backboneSimilar

\[ B(1) = a \]
\[ B(n) \leq c + \max_{k=1\ldots n-1} (B(k) + B(n - k - 1)) \]

How much work are we really doing?
How often do we "visit" each node of t1? At most once, right? And all we do is test whether it has children or not!
Seems like \( cn \) total work
  at least as long as \( a < c \)
  if not, we can increase \( c \) to make it at least as big as \( a \).
Usual well-ordering proof

• Suppose that

\[
\begin{align*}
B(1) &= a \\
B(n) &\leq c + \max_{k=1\ldots n-1} (B(k) + B(n-k-1))
\end{align*}
\]

and \( a \leq c \).

Then I claim that (*) \( B(n) \leq cn \) for \( n = 1, 2, \ldots \).

Let \( S \) be the set of all natural numbers for which (*) is false. Observe that 1 is not in \( S \). Suppose \( S \) nonempty, and we'll arrive at a contradiction.

Let \( h \) be the least element of \( S \) (well-ordering). Then (*) holds for \( n = 1\ldots h-1 \).
Usual well-ordering proof

• Suppose that

\[ B(1) = a \]
\[ B(n) \leq c + \max_{k=1}^{n-1} (B(k) + B(n - k - 1)) \]

and \( a \leq c \). Claim (*) \( B(n) \leq cn \) for \( n = 1, 2, ... \)

Let \( S \) be the set of all natural numbers for which (*) is false. Let \( h \) be the least element of \( S \) (well-ordering). Then (*) holds for \( n = 1 \ldots h-1 \). What's \( B(h) \)? Well,

\[ B(h) \leq c + \max_{k=1}^{h-1} (B(k) + B(h - k - 1)) \]

\[ = c + \max_{k=1}^{h-1} ((ck) + c(h - k - 1)) \]
\[ = c + \max_{k=1}^{h-1} (ck + c(h - 1) - ck) \]
\[ = c + \max_{k=1}^{h-1} (c(h - 1)) = c + c(h - 1) = ch. \]

Contradiction!
Median-finding
Warmup: ceilings

For \( r \in \mathbb{R} \), we have \( \lfloor r \rfloor \geq r \), hence \( a \lfloor r \rfloor \geq ar \) for \( a > 0 \).

For \( k \in \mathbb{N} \), we have \( \left\lfloor \frac{1}{2} k \right\rfloor \geq \frac{k}{2} \) and \( \left\lfloor \frac{1}{5} k \right\rfloor \geq \frac{k}{5} \).

Reason: apply previous result to \( r = \frac{k}{2} \) and \( r = \frac{k}{5} \).

For \( n \in \mathbb{N} \), we have \( \left\lfloor \frac{1}{2} \left\lfloor \frac{n}{5} \right\rfloor \right\rfloor \geq \frac{n}{10} \).

Reason: apply previous result to \( k = \left\lfloor \frac{n}{5} \right\rfloor \) to get \( \left\lfloor \frac{1}{2} \left\lfloor \frac{n}{5} \right\rfloor \right\rfloor \geq \frac{\left\lfloor \frac{n}{5} \right\rfloor}{2} = \frac{1}{2} \left\lfloor \frac{n}{5} \right\rfloor \).

Then apply part 2 to \( k = n \) to get \( \frac{1}{2} \left\lfloor \frac{n}{5} \right\rfloor = \frac{n}{10} \).
• For $n \in \mathbb{N}$, we have $\left\lfloor \frac{1}{2} \left\lceil \frac{n}{5} \right\rceil \right\rfloor \geq \frac{n}{10}$.

• For $n \in \mathbb{N}$, we have $3 \left\lfloor \frac{1}{2} \left\lceil \frac{n}{5} \right\rceil \right\rfloor \geq \frac{3n}{10}$.

• For $n \in \mathbb{N}$, we have $3 \left( \left\lfloor \frac{1}{2} \left\lceil \frac{n}{5} \right\rceil \right\rfloor - 2 \right) \geq \frac{3n}{10} - 6$.
Last facts about ceilings

• For $r \in \mathbb{R}$, we have $\lfloor r \rfloor \leq r + 1$. In particular
• For $n \in \mathbb{N}$, we have $\left\lfloor \frac{n}{5} \right\rfloor \leq \frac{n}{5} + 1$. 
A problem

• Find the (upper) median of a list of n items.
  • Upper median means "if the list has an even number 2s of items, pick the one that's s + 1 from the bottom, rather than s from the bottom"

• Obvious solution: sort, then pick the middle item.
• Seems like more work than is needed.

• Generalize ('strengthen the recursion'): SELECT\(k, S\): find the \(k\)th smallest in a set \(S\) of \(n\) items.
  • Illustrate with sets of numbers, ordered smallest to largest

• MEDIAN\(S\) is now just SELECT\(\left\lceil \frac{n+1}{2} \right\rceil, S\).
A SELECT algorithm (Blum, Floyd, Pratt, Rivest, Tarjan, 1973)

• Input: a nonempty set of \( n \) numbers, and an index \( k, 1 \leq k \leq n \).
• Output: The \( k \)th smallest of the numbers.

1. \text{If } n = 1 \text{ (one item set), return that item.}
2. Divide input into \( \left\lfloor \frac{n}{5} \right\rfloor \) groups of five, and at most one group of \( n \mod 5 \) remaining items.
3. Find the (upper) medians of each of these \( \left\lfloor \frac{n}{5} \right\rfloor \) groups.
4. Find the median \( x \) of these \( \left\lfloor \frac{n}{5} \right\rfloor \) medians (recursively)
5. Partition the input around this median. Let \( p \) be the number of elements on the low side.
   • \text{Low side: all items less than or equal to } x. \text{ High side: items greater than } x.
6. If \( k \leq p \), find the \( k \)th smallest item on the low side; otherwise find the \( k - p \)th smallest item in the high side (recursively)
Group into 5s; median of medians; partition; recur on appropriate piece

Input: 1 5 2 9 8 3 7 4 11 22 27 14 6 21 31 13 12; find 14th-smallest item.

1  3  27  13
5  7  14  12
2  4  6
9  11  21
8  22  31

5  7  13  21
1  5  2  9  8  3  7  4  11  6  12  13

22  27  14  21  31

12 items less than or equal to median of medians; want 14th item. So SELECT(2, upper group), recursively.
A SELECT algorithm (Blum, Floyd, Pratt, Rivest, Tarjan)

• Input: a set of $n$ numbers, and an index $k$, $1 \leq k \leq n$.
• Output: The $k$th smallest of the numbers.

1. Group into 5s; find medians of each (at a cost of $a$ for each); find median of medians
2. Partition around median of medians. Recur.

• Fictitious, experimental analysis.
Suppose that each "part" was no larger than $\frac{3}{4}$ of input. Then we'd have

$$T(n) \leq cn + a \left\lfloor \frac{n}{5} \right\rfloor + T \left\lfloor \frac{n}{3} \right\rfloor + T \left\lfloor \frac{3n}{4} \right\rfloor$$

Replace $a \left\lfloor \frac{n}{5} \right\rfloor$ with similar $\frac{n}{5} = \frac{a}{5} n$, and combine with previous term ($c' = c + a$):

$$T(n) \leq c'n + T \left\lfloor \frac{n}{5} \right\rfloor + T \left\lfloor \frac{3n}{4} \right\rfloor$$
\[ T(n) \leq c'n + T \left( \frac{n}{5} \right) + T \left( \frac{3n}{4} \right) \]

"Guess" \( s = 20c' \); I then claim this looks consistent with \( T(n) \leq sn \) for all \( n \). For ignoring the "ceilings" for a moment, we'd then get

\[
T(n) \leq c'n + s \left( \frac{n}{5} \right) + s \left( \frac{3n}{4} \right)
\]

\[
= c'n + s \left( \frac{19n}{20} \right)
\]

\[
= sn + c'n - \frac{s}{20}n
\]

\[
= sn + c'n - \frac{20c'}{20}n = sn
\]
In practice, it's a little messier than this.

• Warning: Some of the following steps look like magic.
• Carefully crafted to make the algebra as simple as possible.
• Recall from warmup: For $n \in \mathbb{N}$, we have $3 \left( \left\lfloor \frac{1}{2} \left\lfloor \frac{n}{5} \right\rfloor \right\rfloor - 2 \right) \geq \frac{3n}{10} - 6$.
• Critical step: show that in recursive call, the partition piece we recur on is not too big.
  • $\frac{3}{4}n$ was almost too large
  • We'll show it's more like 70%, but with a slight adjustment.
Claim: after partitioning, each "pile" has at least \( \frac{3n}{10} \) numbers in it (almost)
Claim: after partitioning, each "pile" has at least $\frac{3n}{10}$ numbers in it (almost)
Claim: after partitioning, each "pile" has at least $\frac{3n}{10}$ numbers in it (almost)

- "Greater than" pile is no larger than the other
- Contains at least half of the $\left\lfloor \frac{n}{5} \right\rfloor$ medians
- "Greater than" pile is no larger than the other
- Contains at least half of the $\left\lfloor \frac{n}{5} \right\rfloor$ medians
- At least $3 \left( \left\lfloor \frac{n}{5} \right\rfloor - 2 \right) \geq \frac{3n}{10} - 6$ els in "greater than pile"
- At most $\frac{7n}{10} + 6$ els in "≤" pile (or greater-than pile)
Recurrence

- Let $T(n)$ be the max number of operations involved in "Select" on any input of size $n$.
- Group into fives:
  - $cn$
- Find medians of each group:
  - $a[n/5]$
- median of medians:
  - $T[n/5]$
- partition around median element:
  - $bn$ (combine: $c' = c + b$)
- recur on appropriate piece:
  - At most $\frac{7n}{10} + 6$ elts in pile
  - Operation count: $\leq T[\frac{7n}{10} + 6 ]$
- Total: $T(n) \leq c'n + a[n/5] + T[n/5] + T[\frac{7n}{10} + 6 ]$
Algebra

\[ T(n) \leq c'n + a\lfloor n/5 \rfloor + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \]
\[ \leq c'n + a\left( \frac{n}{5} + 1 \right) + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \]
\[ \leq c'n + \frac{a}{5} n + a + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \]
\[ \leq c''n + a + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \] (Note: \( c'' = c' + \frac{a}{5} \))

• Since \( n \) is at least 1, we can write

\[ \leq c''n + an + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \]

\[ = c''''n + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \] (Note: \( c''' = c'' + a \))
Summary

(replacing $c'''$ with $c$)

• For $n \geq 1$,

\[ T(n) \leq cn + T[n/5] + T\left[ \frac{7n}{10} + 6 \right] \]
Algebraic Cleverness

For $n \geq 1, T(n) \leq cn + T[n/5] + T \left[ \frac{7n}{10} + 6 \right]$

1. For $n = 1 \ldots 160$, compute $T(n)$ explicitly, and pick a number $d$ with $T(n) \leq dn$ for $n$ in this range.
   1. Let $d = \max_{n=1,\ldots,160} \frac{T(n)}{n}$, for instance!

2. Pick $s = \max(d, 20c)$. (!)
   1. Because $s \geq 20c$, we have $c \leq \frac{s}{20}$. Also: $d \leq s$

3. I claim that for all $n$, $T(n) \leq sn$.

4. For $n \leq 160$, we have $T(n) \leq dn \leq sn$. (Item 2: $d \leq s$)

5. Still need to handle the case $n > 160$.

6. Why 160? Because it's large enough to make the argument work!
Claim: \( T(n) \leq sn \) for all \( n \)

• Suppose it's false for some minimum value \( k \), but true for all smaller \( n \).
• Then \( k > 160 \) (because we already showed it true for 160 and less). Hence \( \frac{k}{20} > 8 \) (used later).

\[
T(k) \leq ck + T\left[\frac{k}{5}\right] + T\left[\frac{7k}{10} + 6\right]
\]

\[
\leq ck + s\left[\frac{k}{5}\right] + s\left[\frac{7k}{10} + 6\right]
\]

\[
\leq ck + s\left(\frac{k}{5} + 1\right) + s\left(\frac{7k}{10} + 6 + 1\right)
\]

\[
= ck + s\frac{k}{5} + s + s\frac{7k}{10} + 7s
\]

\[
= ck + s\frac{9k}{10} + 8s
\]

\[
\leq \frac{s}{20}k + s\frac{18k}{20} + 8s
\]

\[
= s\frac{19k}{20} + 8s
\]

\[
= sk + 8s - \frac{1}{20}sk
\]

\[
= sk + s\left(8 - \frac{k}{20}\right) \quad [By \ note \ above, \ \frac{k}{20} > 8, \ so \ 0 > 8 - \frac{k}{20}]
\]

\[
\leq sk
\]

Contradiction! Hence claim is true for all \( n \).
Why piles of five?

• If you try more or fewer, the sum of $\frac{n}{5}$ and $\frac{7n}{10}$ ends up changing to something...a bit larger than 1 instead of a bit less than 1
• So 5 is a "sweet spot" for this algorithm!
Surprising simpler algorithm

- RandSelect(k, S)
  - Pick a random item $x$ in your set, $S$
  - Partition into set $L$ of numbers less than $x$, and set $G$ of those greater than $x$
  - If $L$ has at least $k$ items: RandSelect($k$, $L$)
  - If $L$ has $k-1$ items: return $x$
  - Otherwise, RandSelect($k - \text{size}(L) - 1$, $G$)

- Works in "expected linear time" because on average, the size of the larger partition is $\frac{3}{4}$ size of the set. Work is (roughly)

\[
n + \frac{3}{4}n + \left(\frac{3}{4}\right)^2 n + \ldots = n \frac{1}{1 - \frac{3}{4}} = n \frac{1}{\frac{1}{4}} = 4n.
\]
Big idea! (More in CS18)

• Randomized algorithms are often simpler than deterministic ones
• Deep philosophical question: why does adding a stream of randomness make tasks easier?