⚡Show & Tell⚡

Solving things with bisection

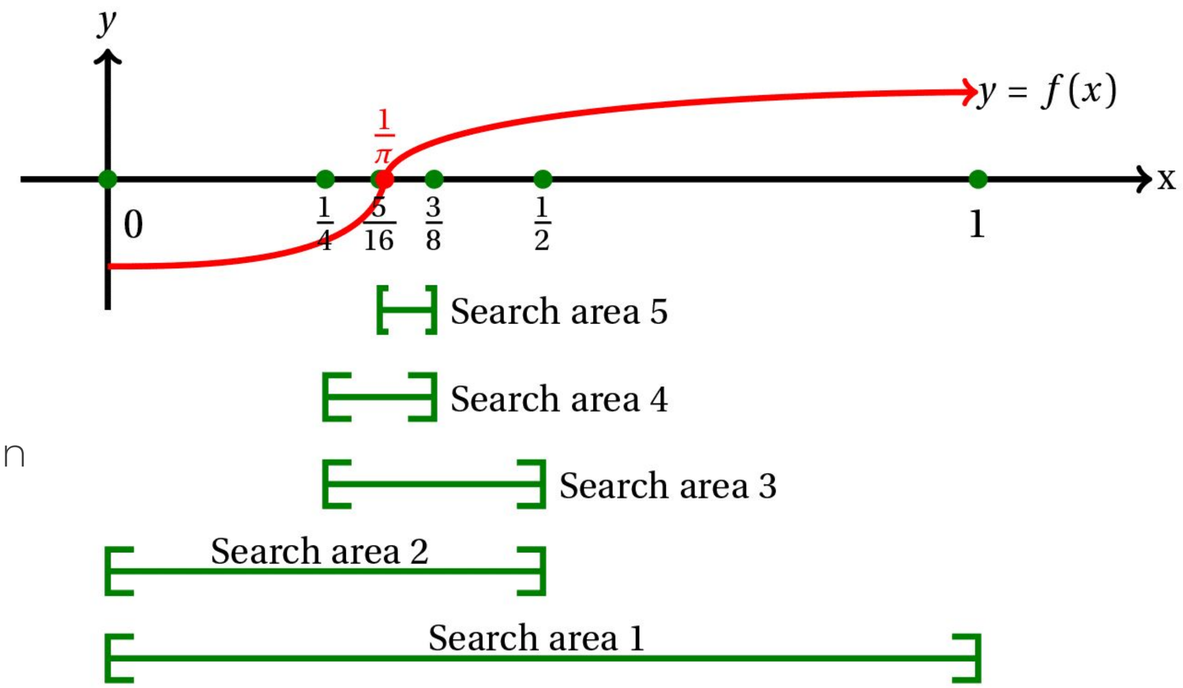
Mathematical analysis

Bisection method

- Root-finding method *f(x) = 0* for real numbers *x*
- Applies to continuous function defined on interval *I =[a, b]*
- Bolzano's theorem
  - Two values *f(a)* and *f(b)* with opposite signs has a root

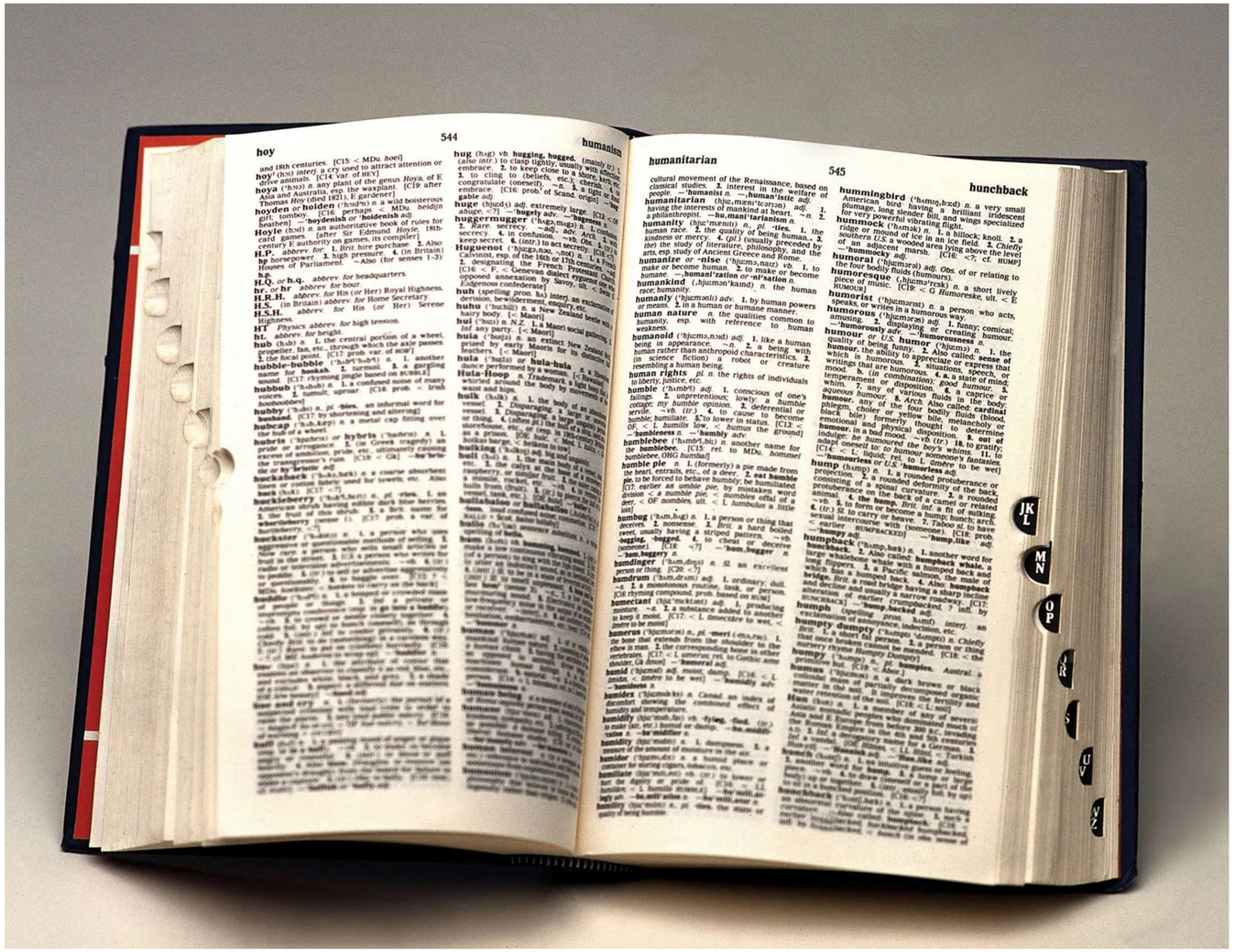
- Repeatedly bisecting the defined interval
- And selecting of subinterval in which function changes sign
- Gives us a rough approximation of the root

- Solving equations in the real numbers



In the wild

Usage of bisection



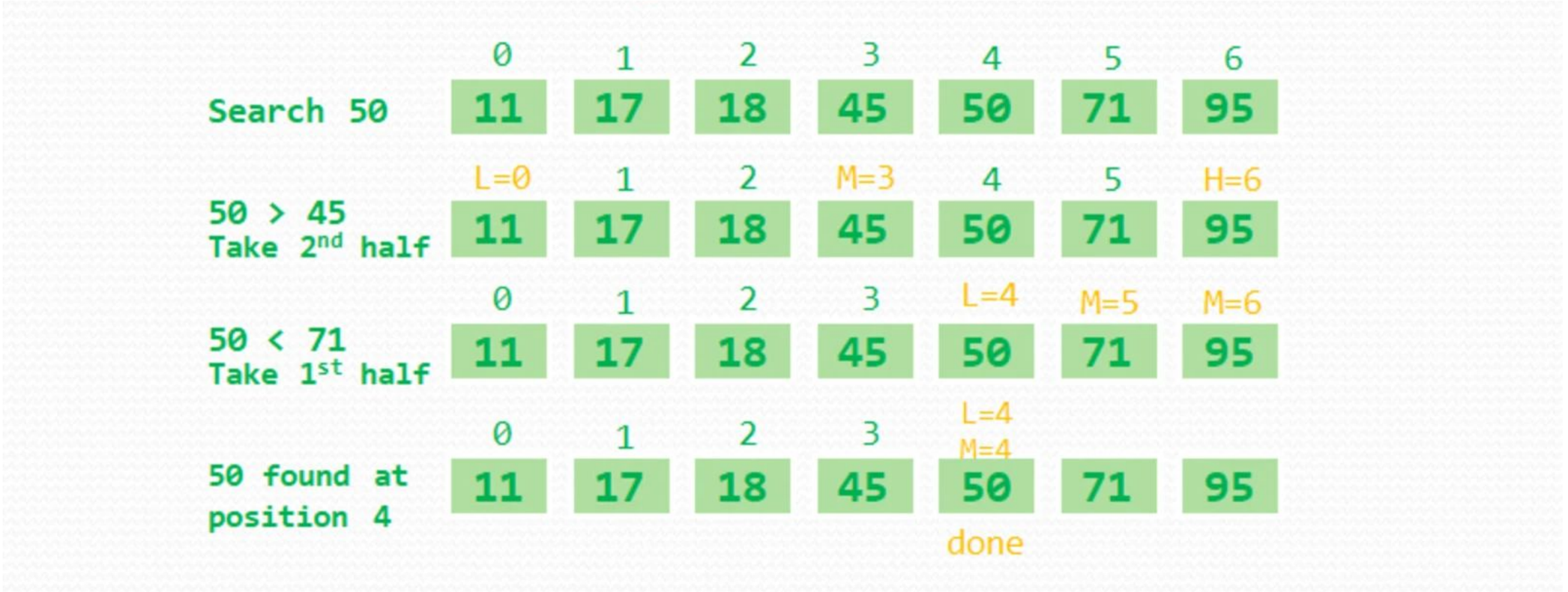
In the wild

Usage of bisection

Guess the number between 0 and 100

Computer Science

Bisection by hand



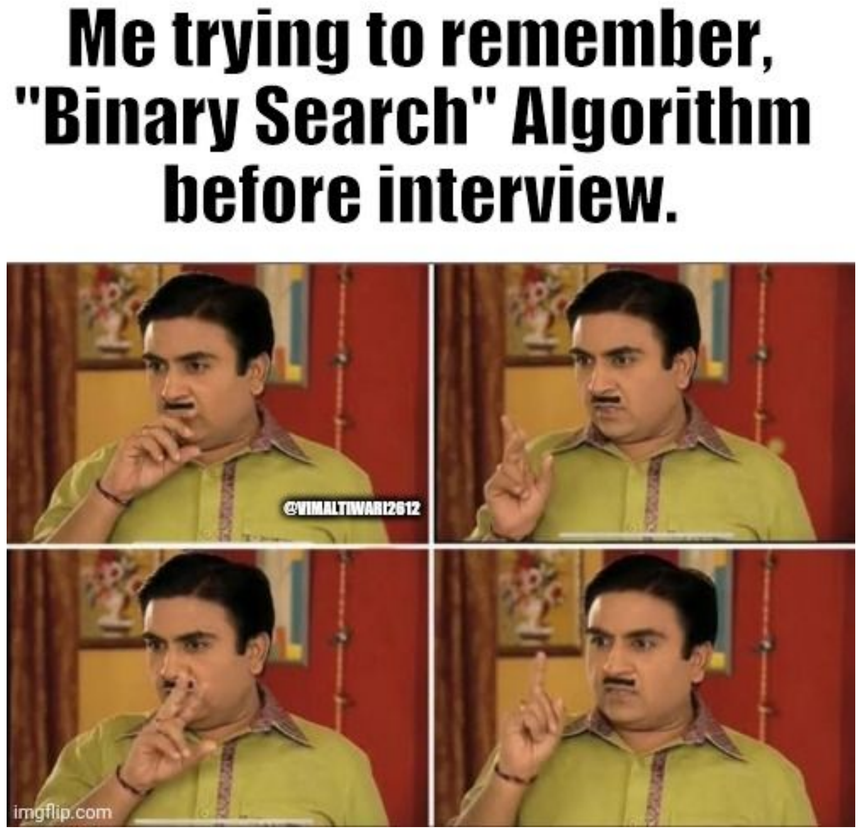
Computer Science

Binary search

- Bisection search / binary search / logarithmic search
- Find the position within a **sorted** array

Performance:

- Worst-case: *O(log n)*
- Best-case: *O(1)*
- Average: *O(log n)*
- Worst-case space complexity: *O(1)*



Although the basic idea of binary search is comparatively straightforward, the details can be surprisingly tricky

Donald Knuth, professor at Stanford University

Git bisect command

NAME

git-bisect - Use binary search to find the commit that introduced a bug

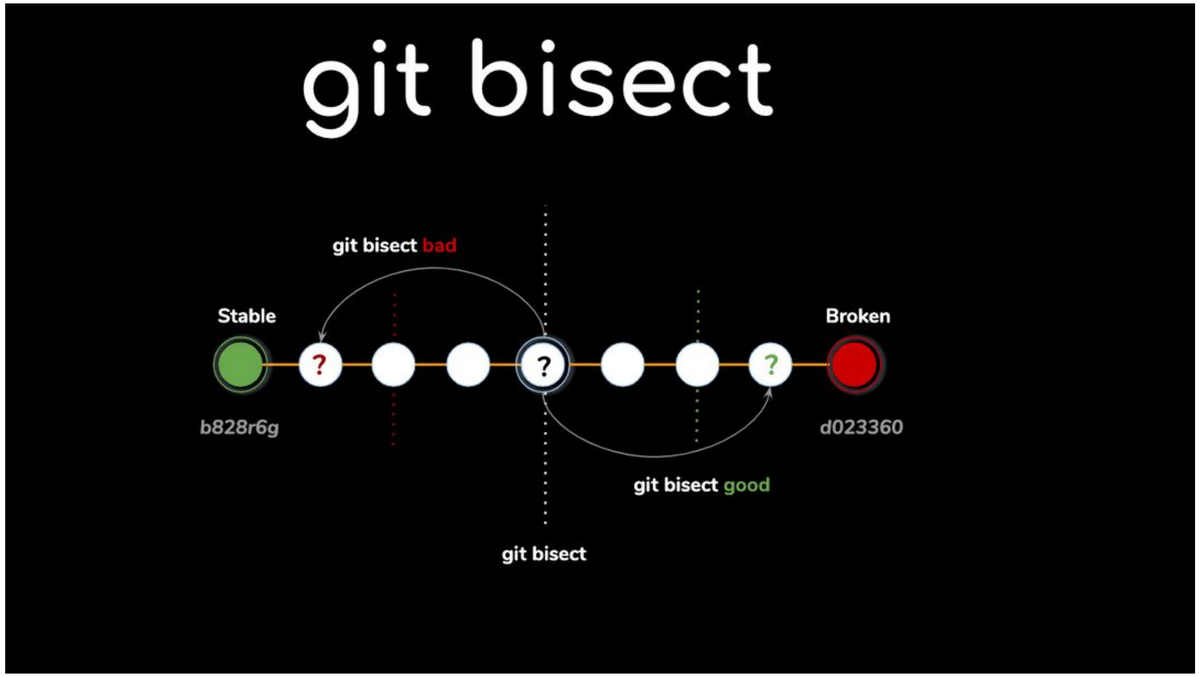
SYNOPSIS

```
git bisect <subcommand> <options>
```

DESCRIPTION

The command takes various subcommands, and different options depending on the subcommand:

```
git bisect start [--term-{new,bad}=<term> --term-{old,good}=<term>]
          [--no-checkout] [--first-parent] [<bad> [<good>...]] [--] [<paths>
git bisect (bad|new|<term-new>) [<rev>]
git bisect (good|old|<term-old>) [<rev>...]
git bisect terms [--term-good | --term-bad]
git bisect skip [(<rev>|<range>)...]
git bisect reset [<commit>]
git bisect (visualize|view)
git bisect replay <logfile>
git bisect log
git bisect run <cmd>...
git bisect help
```



Binary-schminary

Why bother?

Number of commits	Steps to check
10	$\log_2(10) \sim 4$
100	~ 7
1k	~ 10
10k	~ 14
100k	~ 17
1M	~ 20

Demo

Questions
and feedback

<footer>10</footer>