

Binary Search

...

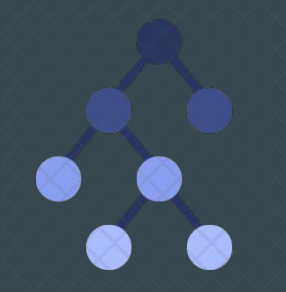
Julia
Pamela

What is Binary Search?



- Algorithm that efficiently searches x within a sorted array.
- Better than linear search in terms of time complexity.

How does it work?



Low

High


$$\text{Mid} = (\text{min} + \text{max}) // 2$$




Challenges

- In the first iteration implemented **less efficient version**.
- After research implemented the **best practice approach**.
- Examples:
 - **Abstract problem**;
 - Implementation used within **GIT** version control system.

Chocolate Bar Example

sweetness =
sum()
= 91

friends =
count()
= 11

We have a chocolate bar that consists of some chunks. **Each chunk has its own sweetness** given by the array **sweetness**.

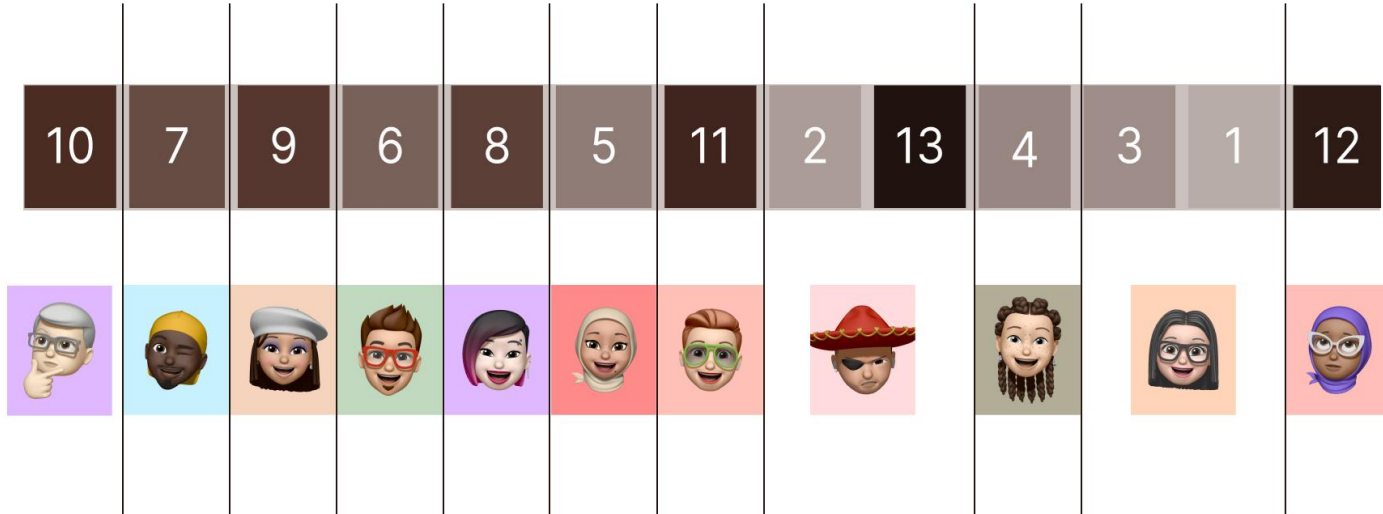
We want to share the chocolate with **11 group mates**, so we start by cutting the chocolate bar into **10 pieces**.

Scenario: Cutting a Chocolate Bar



- We will divide the pieces **randomly**.
- This way someone will end up with a piece with the **minimum total sweetness**.
- To make a fair cut we **maximize the sweetness of such a piece**.
- So we search for **maximum available total sweetness for a piece with minimal sweetness**.

maximum available total sweetness



≥ 4

GIT BISECT EXAMPLE


```
O:0:git-bisect-example - "secretariat"
ugzuzg@secretariat (master) % git bisect start
status: waiting for both good and bad commits
ugzuzg@secretariat (master) % git bisect bad
status: waiting for good commit(s), bad commit known
ugzuzg@secretariat (master) %
```

```
(master) % git bisect start
both good and bad commits
(master) % git bisect bad
good commit(s), bad commit known
```

MARK COMMIT AS BAD

Header

[Nav Item 1](#)

[Nav Item 2](#)

[Nav Item 3](#)

[Nav Item 4](#)

[Nav Item 5](#)

Page Title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Block	Block	Block	Block
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

```
O:0:git-bisect-example - "secretariat"
ugzuzg@secretariat (master) % git bisect start
status: waiting for both good and bad commits
ugzuzg@secretariat (master) % git bisect bad
status: waiting for good commit(s), bad commit known
ugzuzg@secretariat (master) % git checkout 8529499
Note: switching to '8529499'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to do so (now or later) by using

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting

```
HEAD is now at 8529499 adding
ugzuzg@secretariat (8529499)
```

```
Status: waiting for good commit(s), bad commit known
ugzuzg@secretariat (master) % git checkout 8529499
Note: switching to '8529499'.
```

SWITCH TO LAST KNOWN GOOD COMMIT

```
[0] O:git-bisect-example*
```

```
"secretariat" 21:50 09-aug-23
```

Git Bisect Example

file:///home/ugzuzg/gh/EvanLovely/git-bisect-example/index

Header

- [Nav Item 1](#)
- [Nav Item 2](#)
- [Nav Item 3](#)

```
O:0:git-bisect-example - "secretariat"
ugzuzg@secretariat (master) % git bisect start
status: waiting for both good and bad commits
ugzuzg@secretariat (master) % git bisect bad
status: waiting for good commit(s), bad commit known
ugzuzg@secretariat (master) % git checkout 8529499
Note: switching to '8529499'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 8529499 adding basic header styles

```
ugzuzg@secretariat (8529499) % git bisect good
Bisecting: 4 revisions left to test after this (roughly 2 steps)
[498b7a4057ee9adc7244f038f2ff04044fb95e0d] laying out header nav
ugzuzg@secretariat (498b7a4) %
```

HEAD is now at 8529499 adding basic header styles

```
ugzuzg@secretariat (8529499) % git bisect good
Bisecting: 4 revisions left to test after this (roughly 2 steps)
[498b7a4057ee9adc7244f038f2ff04044fb95e0d] laying out header nav
ugzuzg@secretariat (498b7a4) %
```

4 COMMITS LEFT TO TEST WITHIN ABOUT 2 STEPS OF SEARCH

Git Bisect Example

file:///home/ugzuzg/gh/EvanLovely/git-bisect-example/index

Header

- [Nav Item 1](#)
- [Nav Item 2](#)
- [Nav Item 3](#)
- [Nav Item 4](#)
- [Nav Item 5](#)



Possible improvements:

Create a library for binary search function and pass to it the search space and the function of evaluating the current mid so that it can be applied to different tasks.