ClojureScript,

or How to Run LISP Everywhere


cognitect


Java™


JS

Always be transpiling

- CoffeeScript
- TypeScript
- asm.js
- Dart
- ES2015

Data Structures & Interactivity

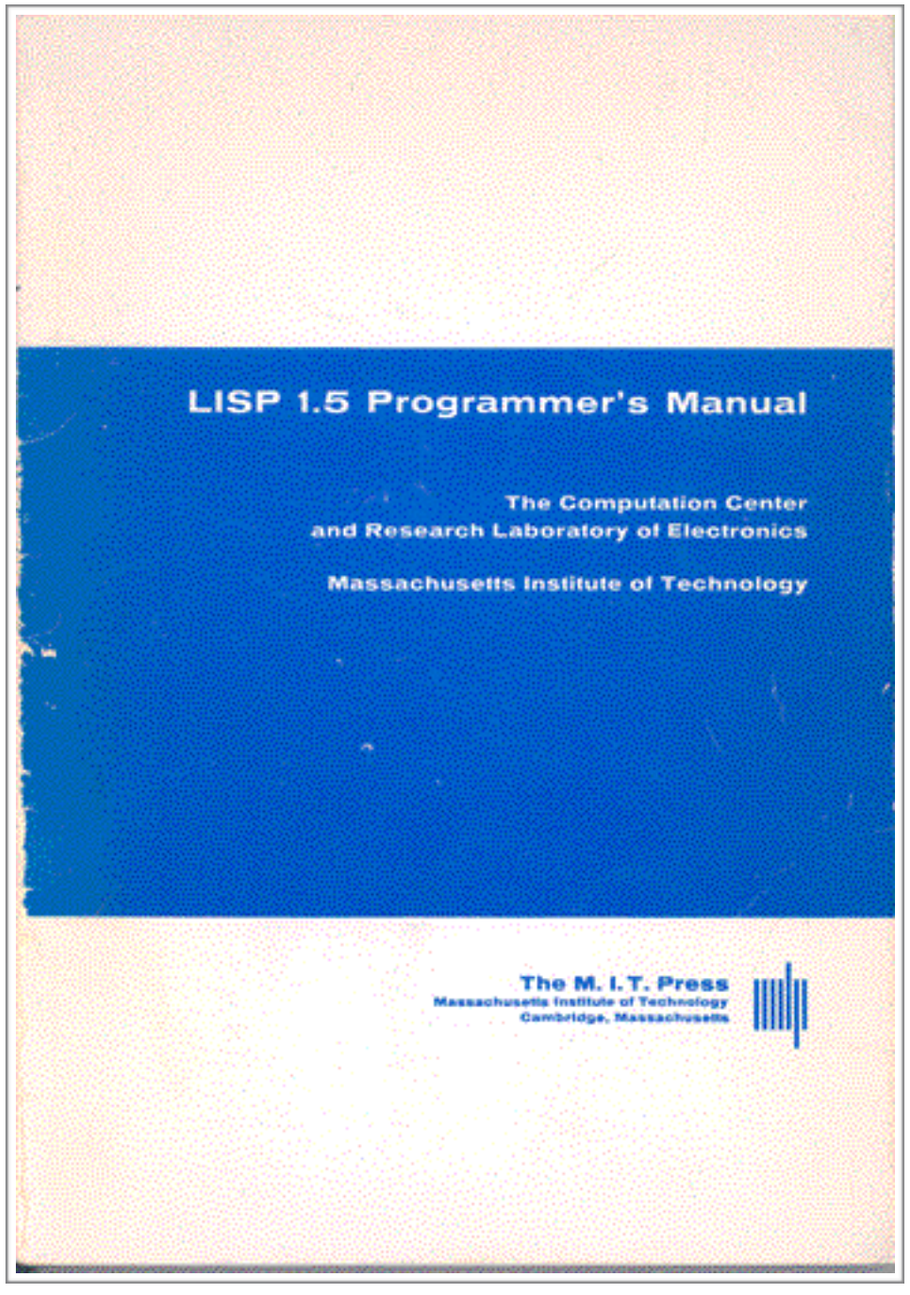
The Dream Machine.

J. C. R. Licklider and the Revolution That Made Computing Personal.

M. Mitchell Waldrop.

author of Complexity.

"Waldrop's account of [Licklider's] and many others' world-transforming contributions is compelling."
—John Allen Paulos, The New York Times Book Review


LISP 1.5 Programmer's Manual
The Computation Center
and Research Laboratory of Electronics
Massachusetts Institute of Technology
The M. I. T. Press
Massachusetts Institute of Technology
Cambridge, Massachusetts

evalquote is defined by using two main functions, called eval and apply. apply handles a function and its arguments, while eval handles forms. Each of these functions also has another argument that is used as an association list for storing the values of bound variables and function names.

```
evalquote[fn;x] = apply[fn;x;NIL]
```

where

```
apply[fn;x;a] =
    [atom[fn] → [eq[fn;CAR] → caar[x];
                 eq[fn;CDR] → cdar[x];
                 eq[fn;CONS] → cons[car[x];cadr[x]];
                 eq[fn;ATOM] → atom[car[x]];
                 eq[fn;EQ] → eq[car[x];cadr[x]];
                 T → apply[eval[fn;a];x;a]];
     eq[car[fn];LAMBDA] → eval[caddr[fn];pairlis[cadr[fn];x;a]];
     eq[car[fn];LABEL] → apply[caddr[fn];x;cons[cons[cadr[fn];
                                                caddr[fn]];a]]]
eval[e;a] = [atom[e] → cdr[assoc[e;a]];
     atom[car[e]] →
          [eq[car[e],QUOTE] → cadr[e];
           eq[car[e];COND] → evcon[cdr[e];a];
           T → apply[car[e];evlis[cdr[e];a];a]];
     T → apply[car[e];evlis[cdr[e];a];a]]
```

pairlis and assoc have been previously defined.

```
evcon[c;a] = [eval[caar[c];a] → eval[cadar[c];a];
              T → evcon[cdr[c];a]]
```

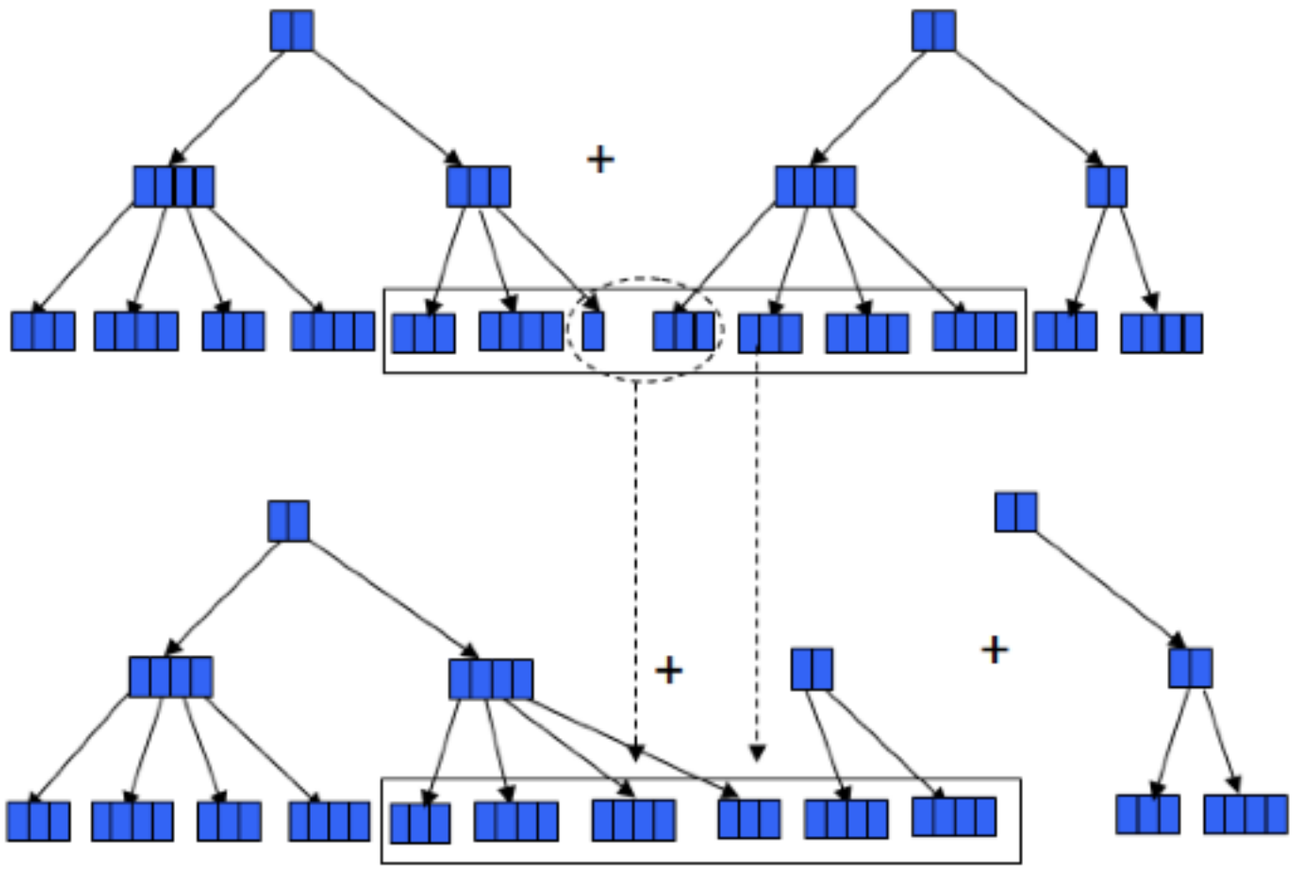
and

```
evlis[m;a] = [null[m] → NIL;
              T → cons[eval[car[m];a];evlis[cdr[m];a]]]
```

ClojureScript

- 45 months old
- >4000 GitHub stars
- 104 contributors
- Users: Consumer Reports, eBay, Prismatic, Reuters ...

- Runs on Rhino & Nashorn
- Runs on browsers back to Internet Explorer 6
- Runs on Node.js & JavaScriptCore (iOS)

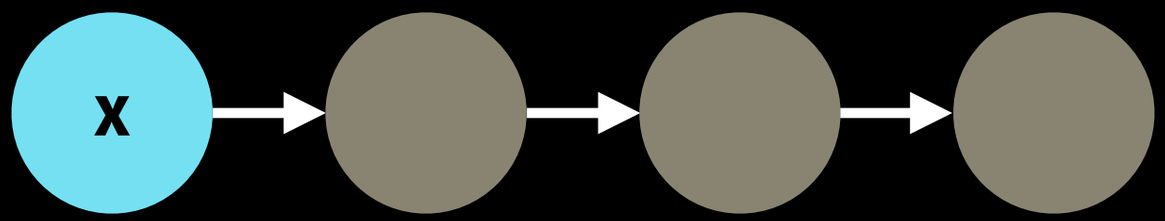



HACKER
SCHOOL
2012

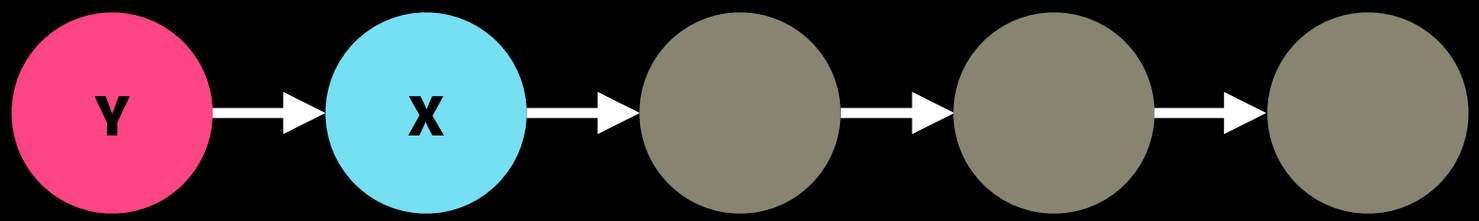
Functional Programming and Data

- immutable values, not mutable objects
- “change” returns a new value, leaving the old one unmodified
- they’re persistent
- they’re fast

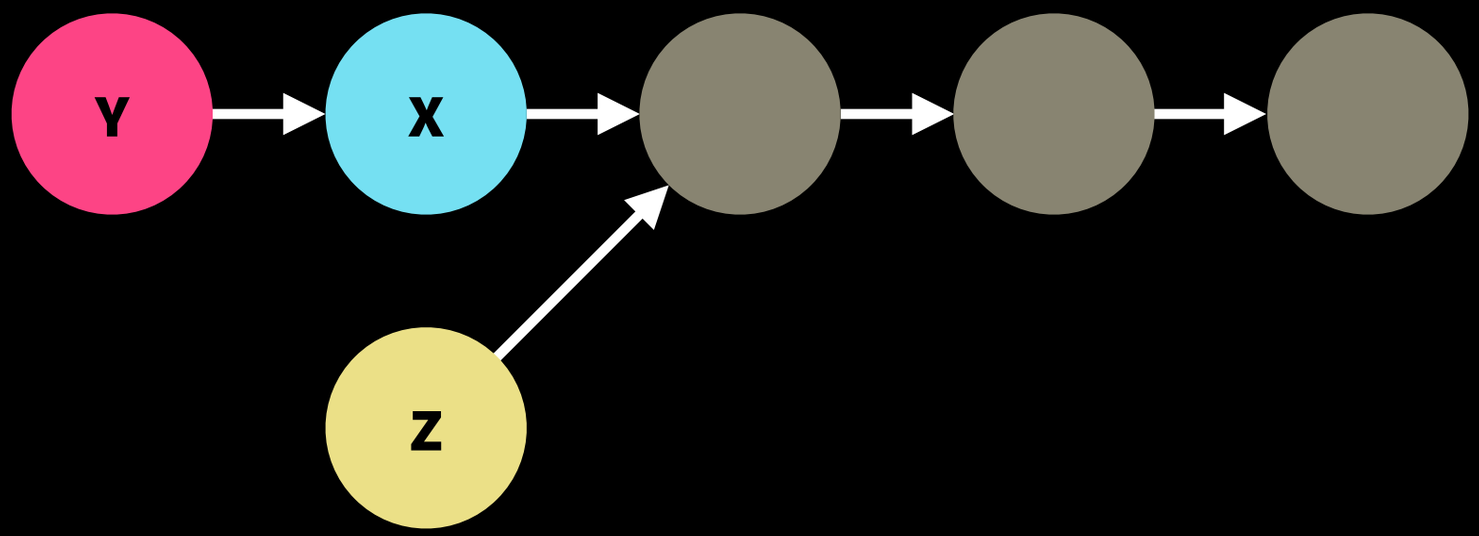
Simple example: Linked List



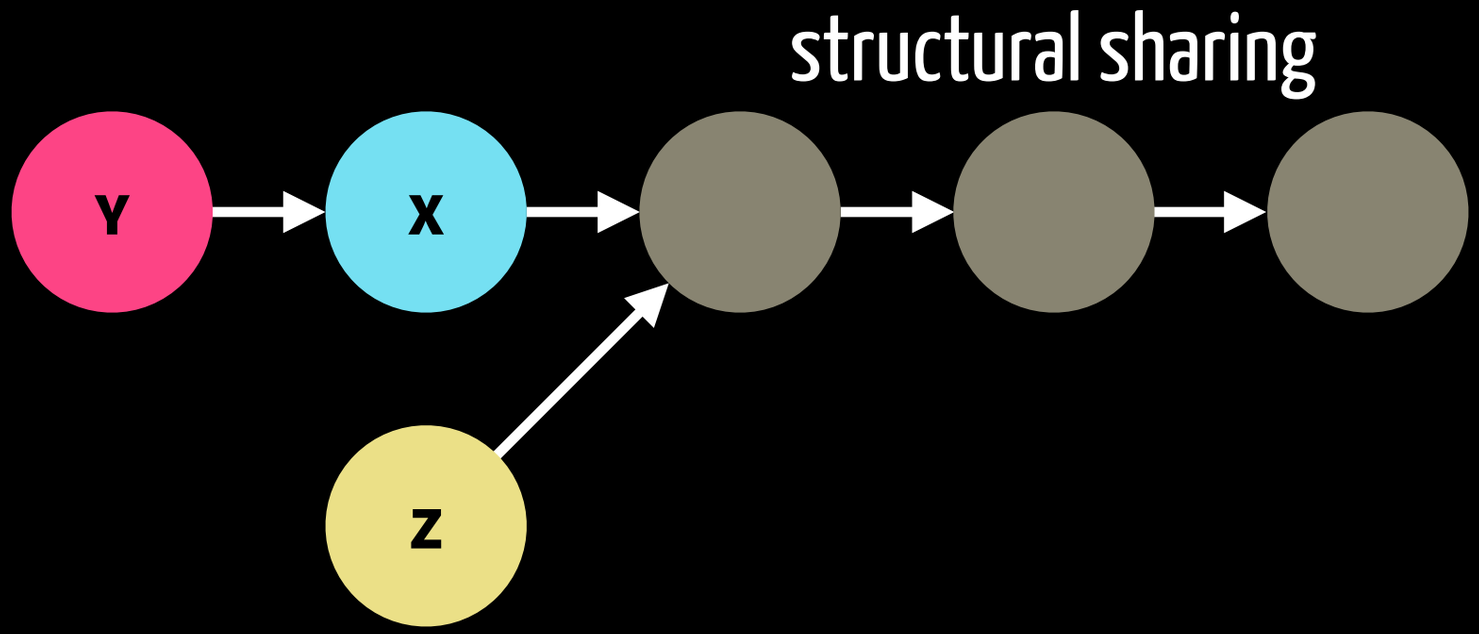
Simple example: Linked List



Simple example: Linked List



Simple example: Linked List



Sharing structure

- space efficiency
- computational efficiency – avoids copying

Phil Bagwell

- Array Mapped Trie
- Hash Array Mapped Trie

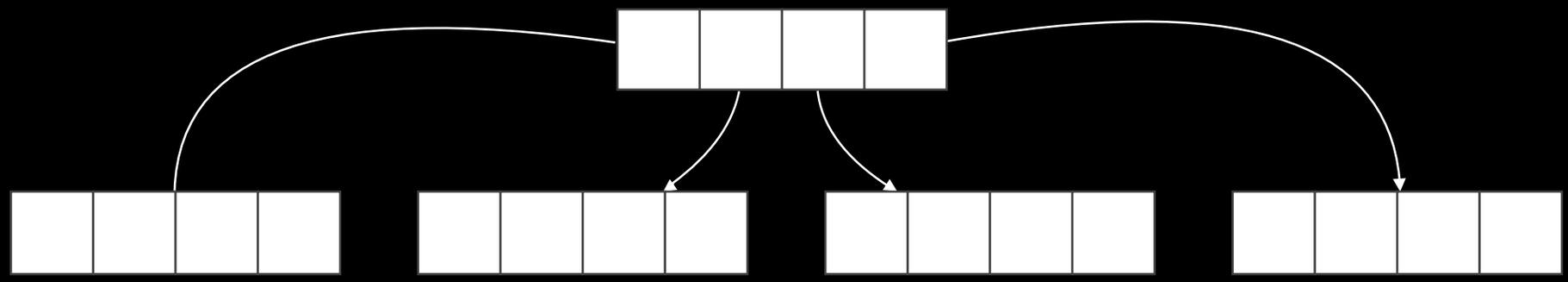
Bitmapped Vector Trie

- data lives in the leaves
- e.g. prefix tree used for string lookup
- bitwise trie

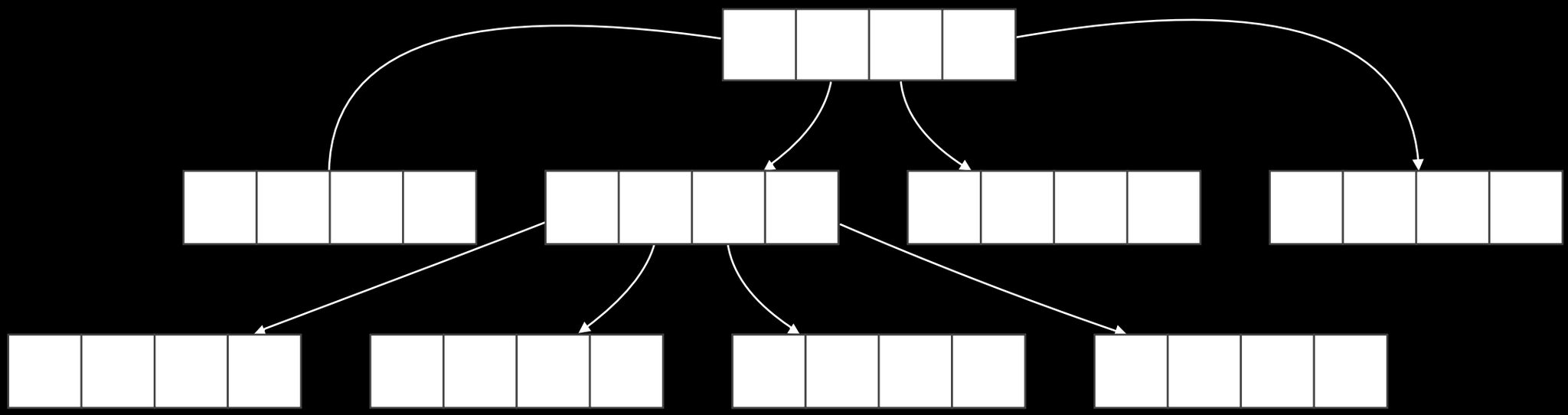
Persistent Vector

Persistent Vector

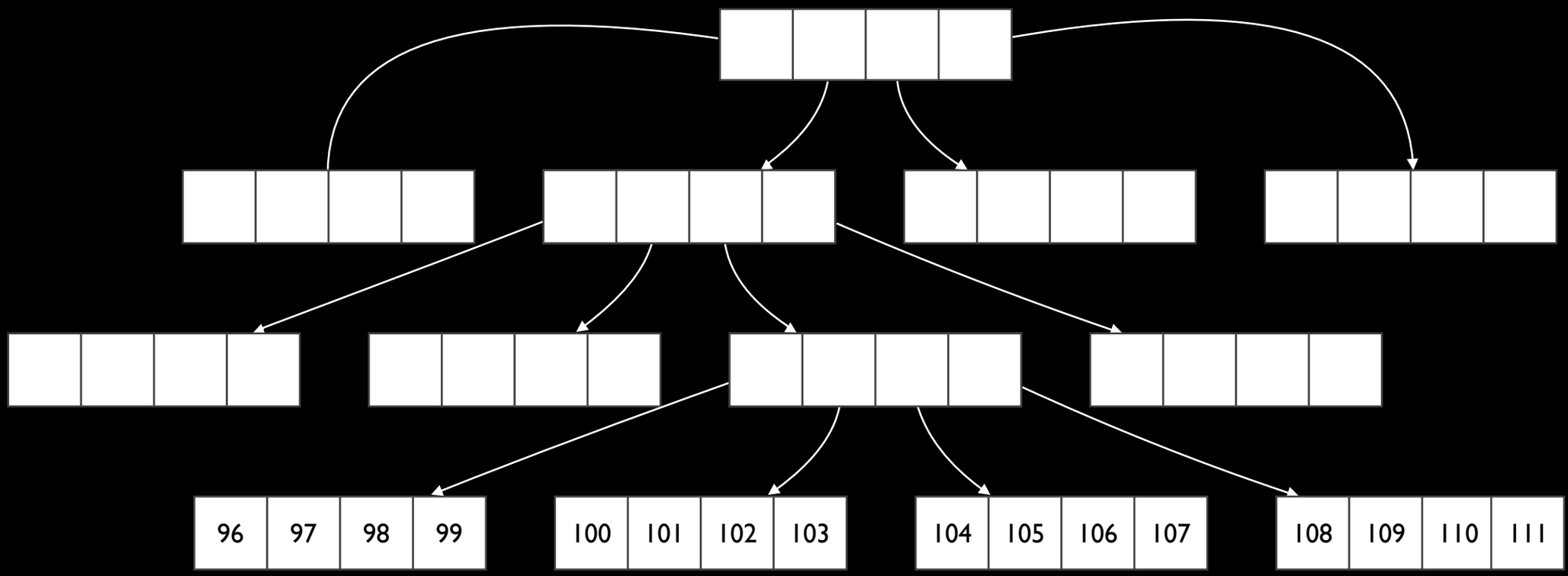
Persistent Vector



Persistent Vector



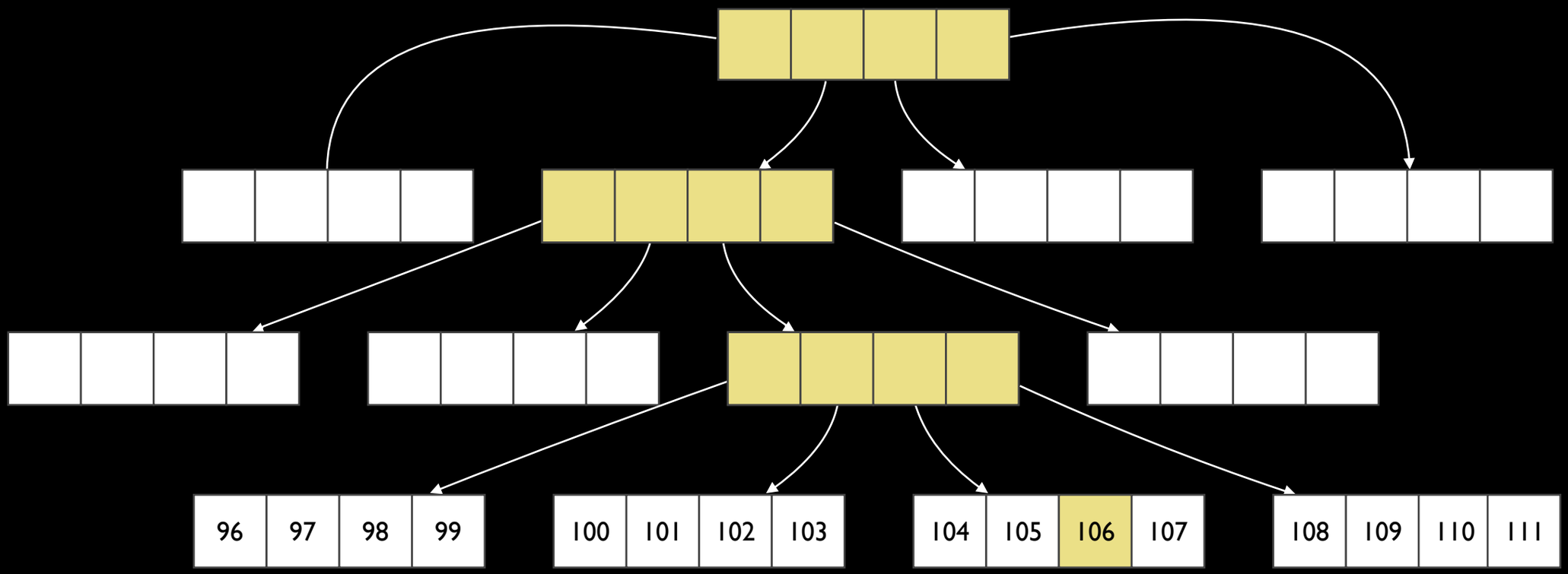
Persistent Vector



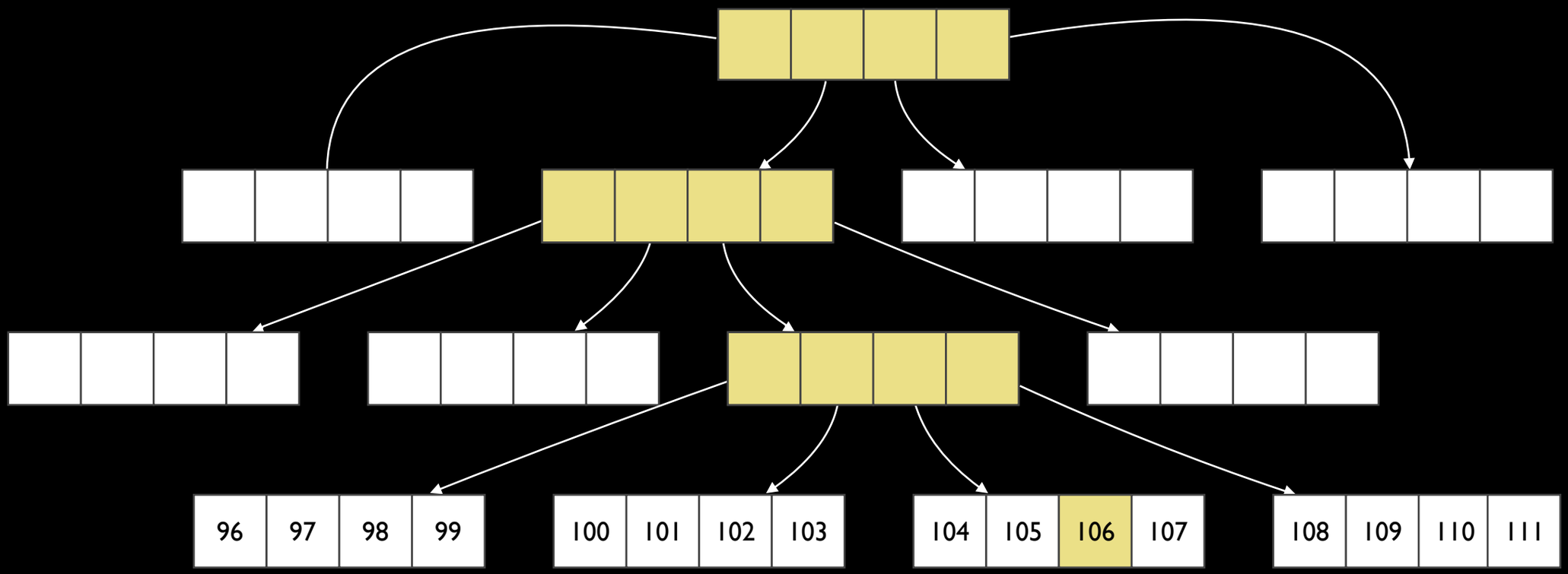
Persistent Vector

`getindex`

Persistent Vector

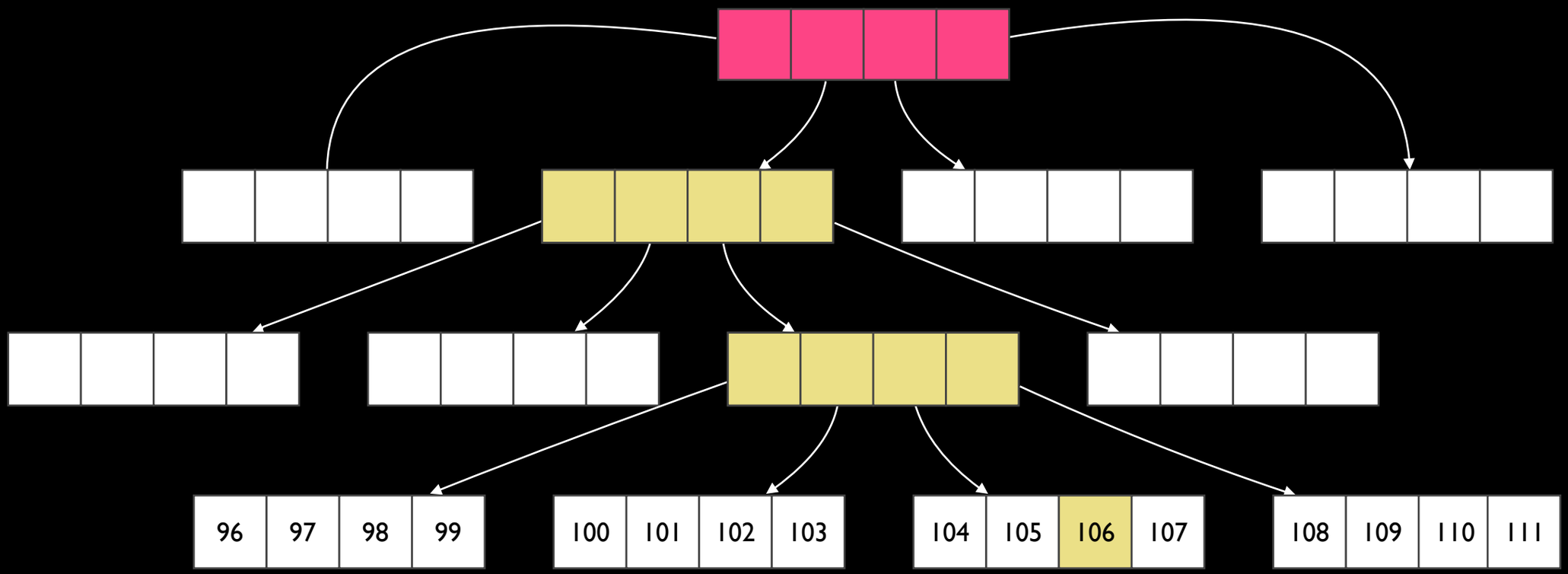


Persistent Vector



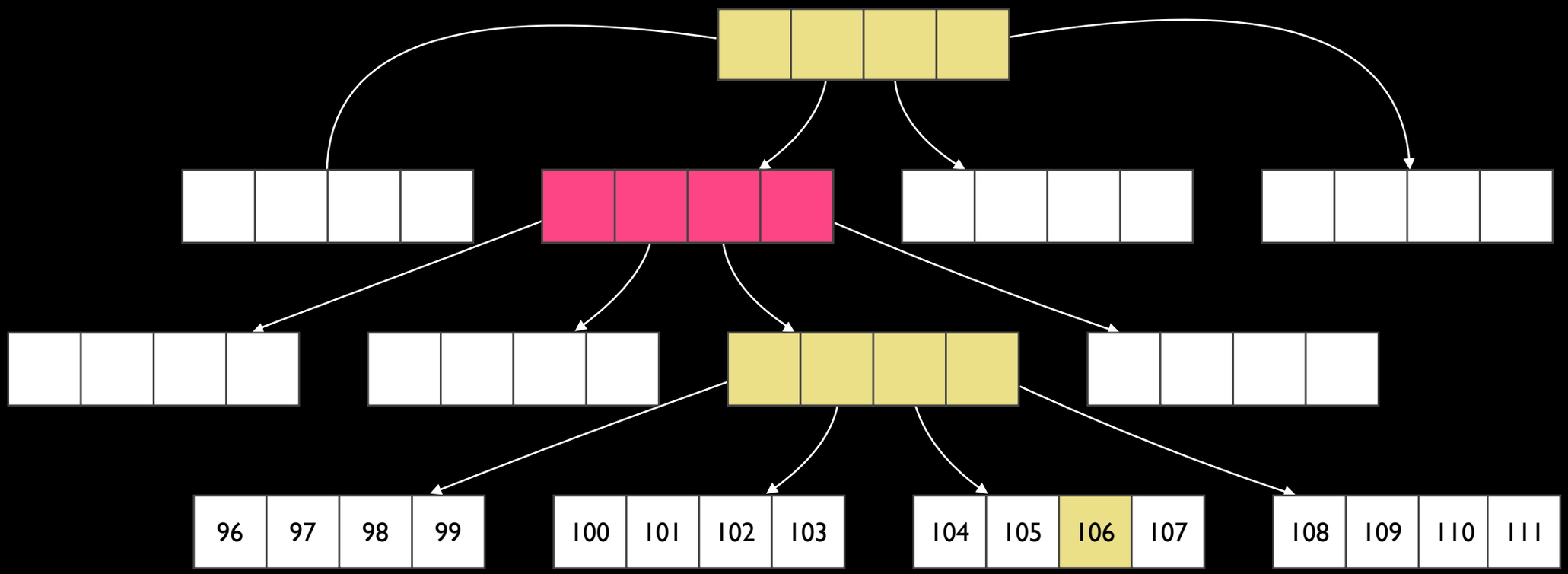
0b01101010

Persistent Vector



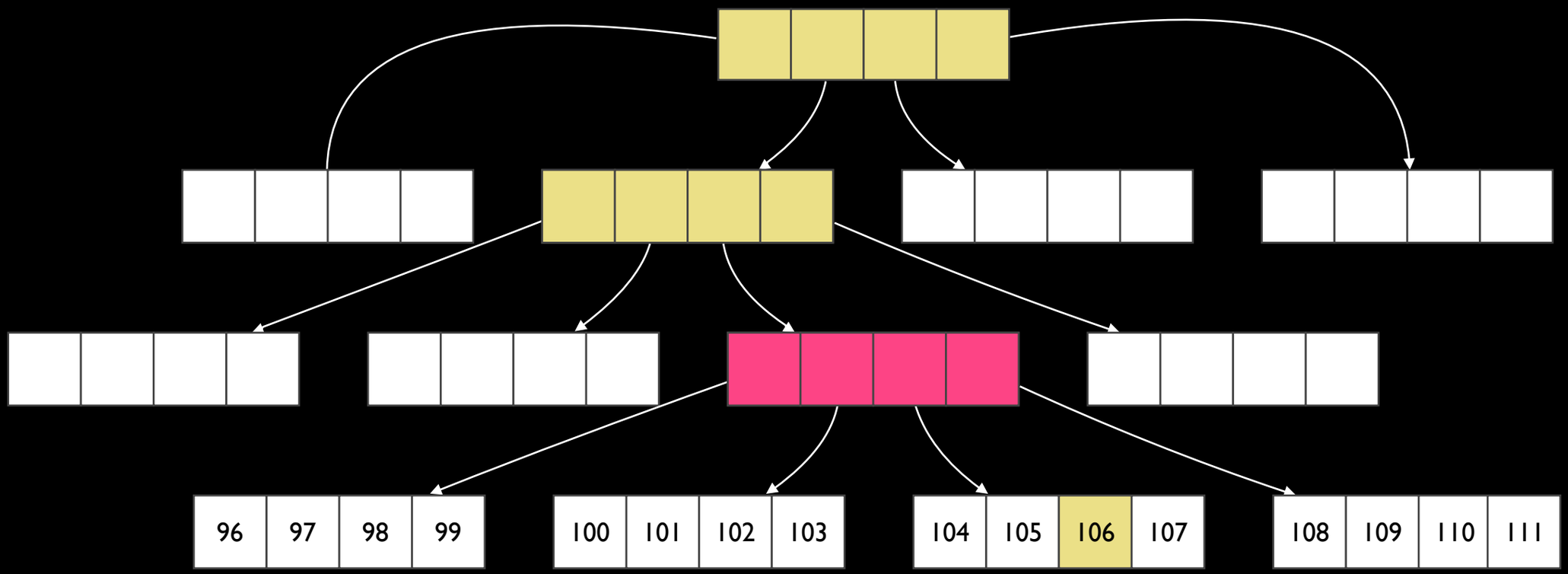
0b01101010

Persistent Vector



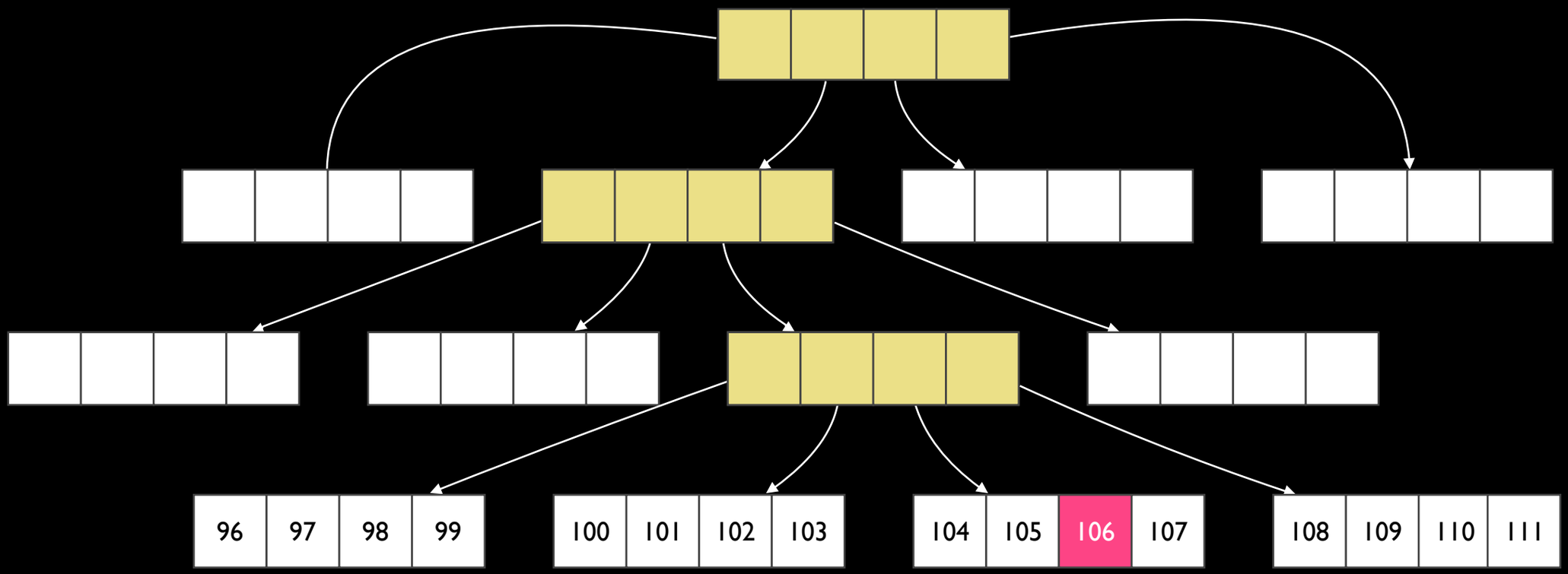
0b01101010

Persistent Vector



0b01101010

Persistent Vector

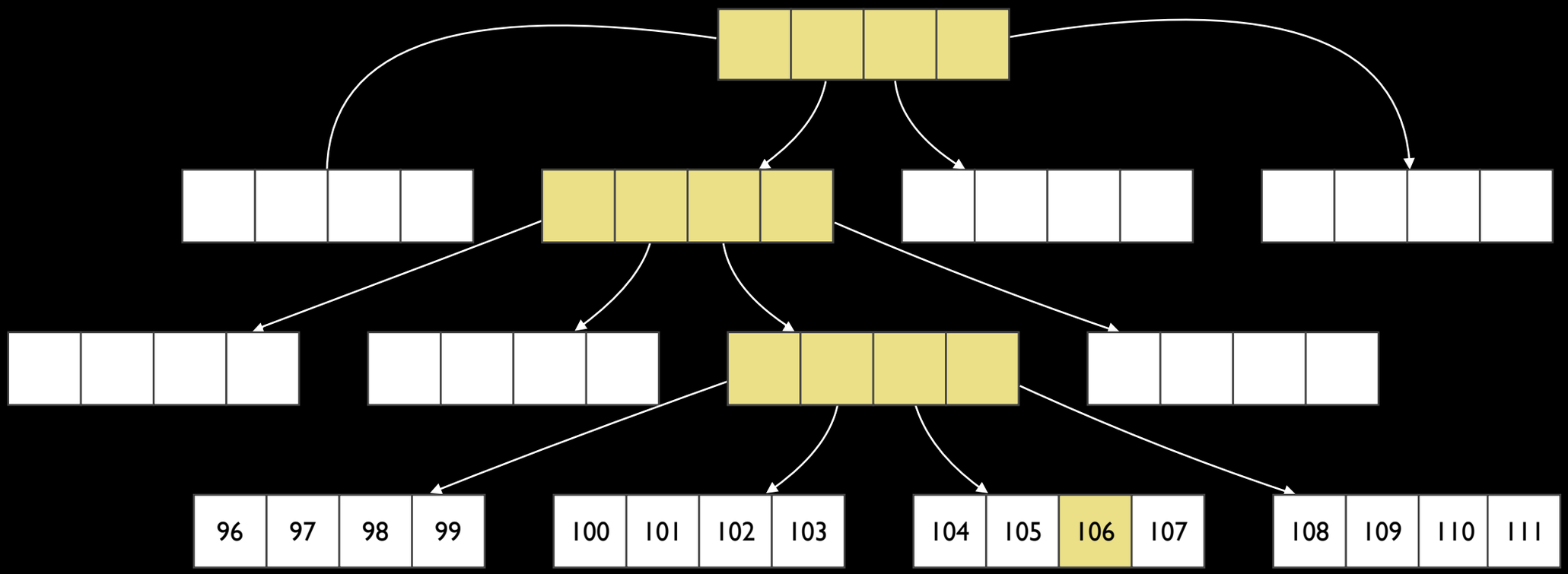


0b01101010

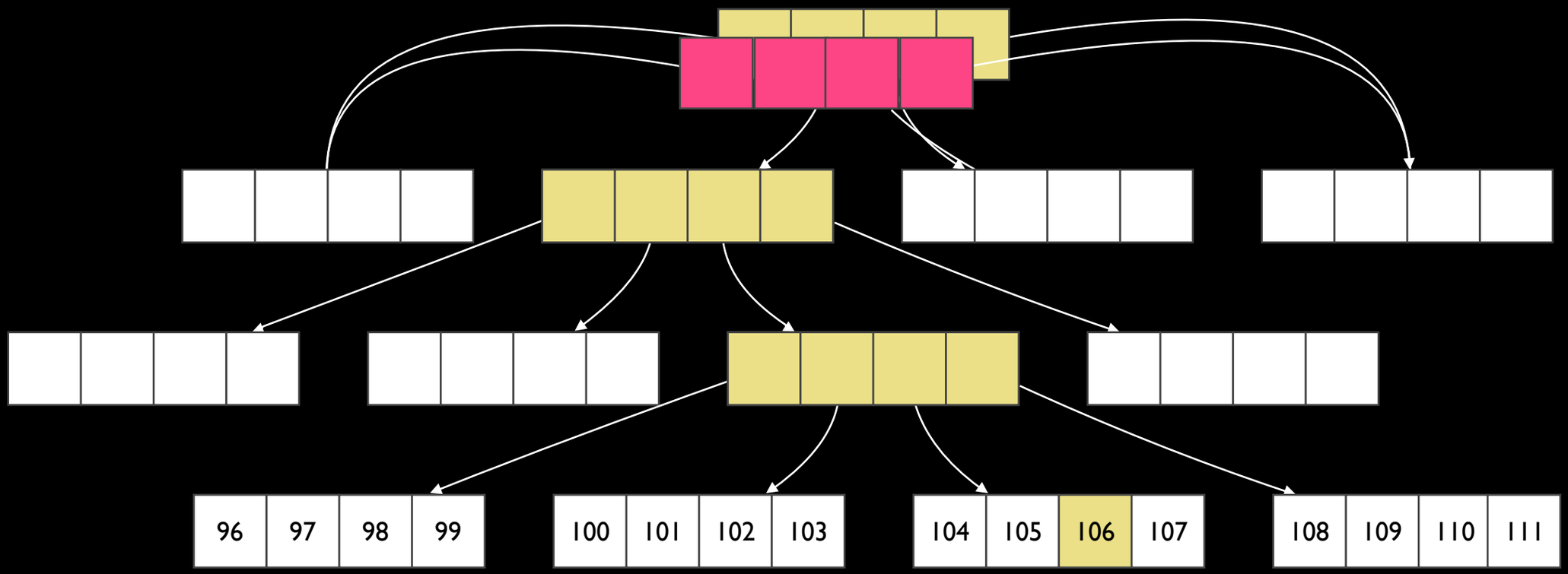
Persistent Vector

assoc

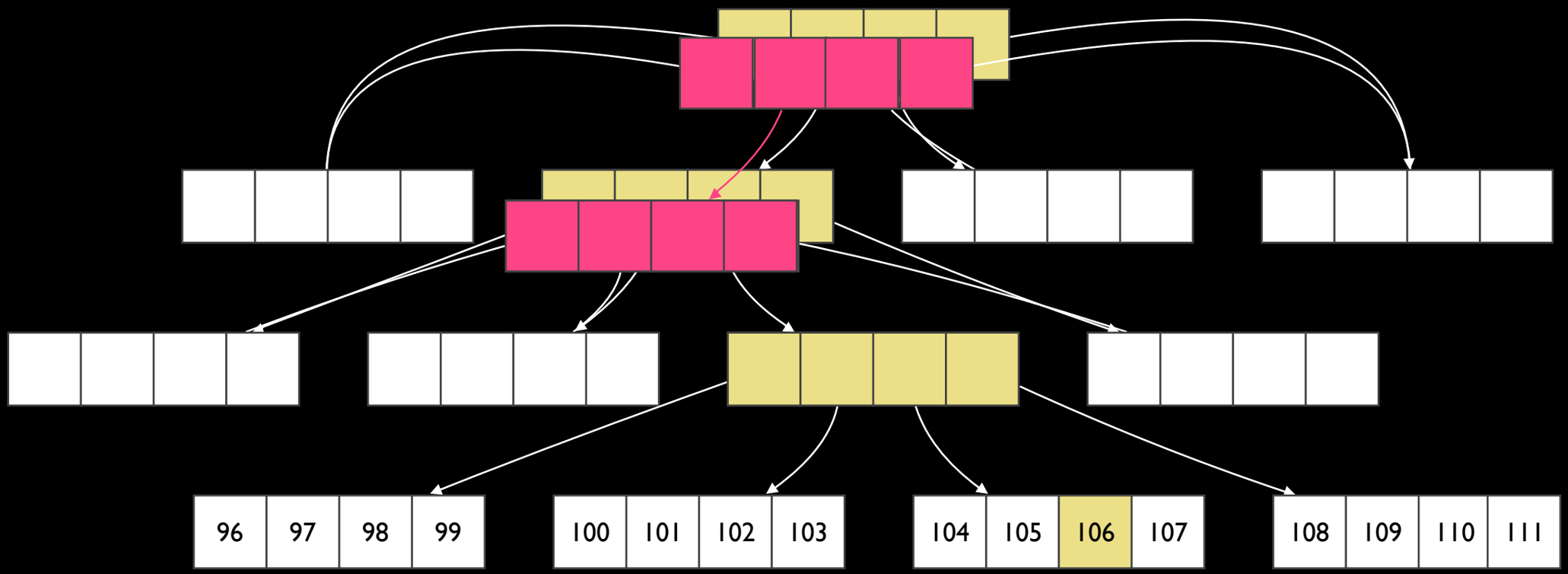
Persistent Vector



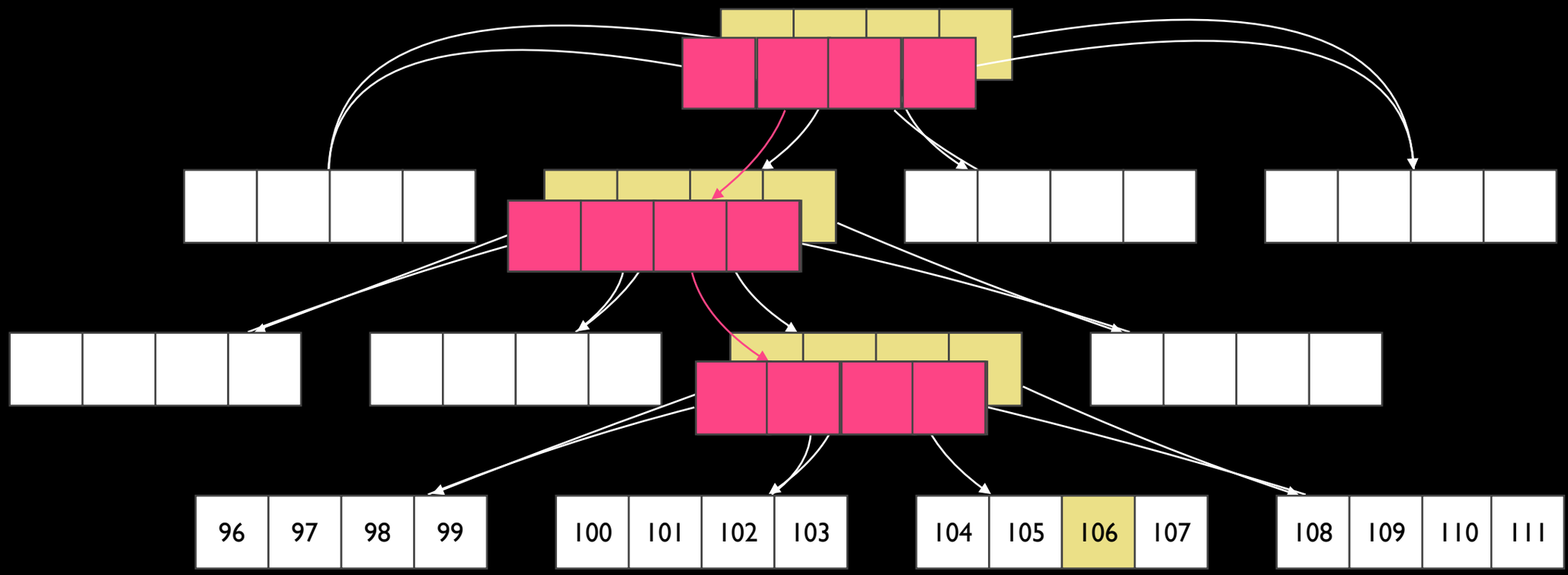
Persistent Vector



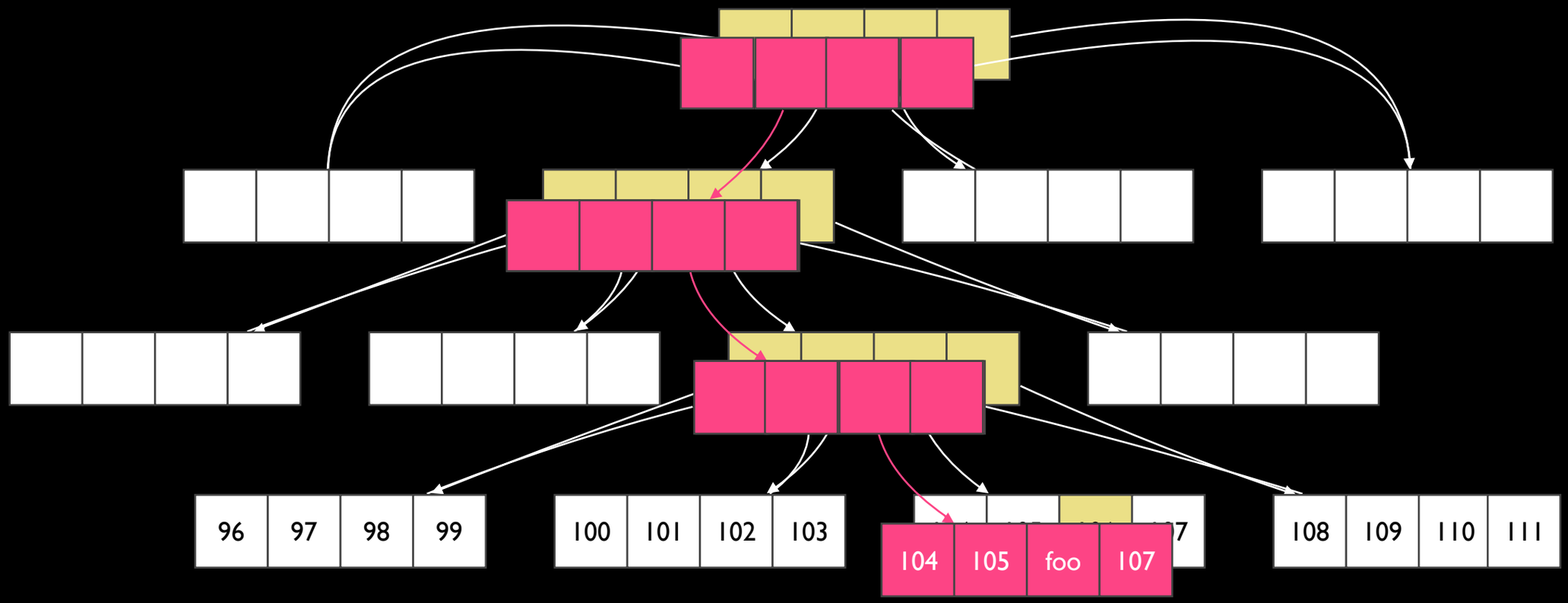
Persistent Vector



Persistent Vector



Persistent Vector



Persistent Vector

Length 4 internal vectors?

Persistent Vector

32

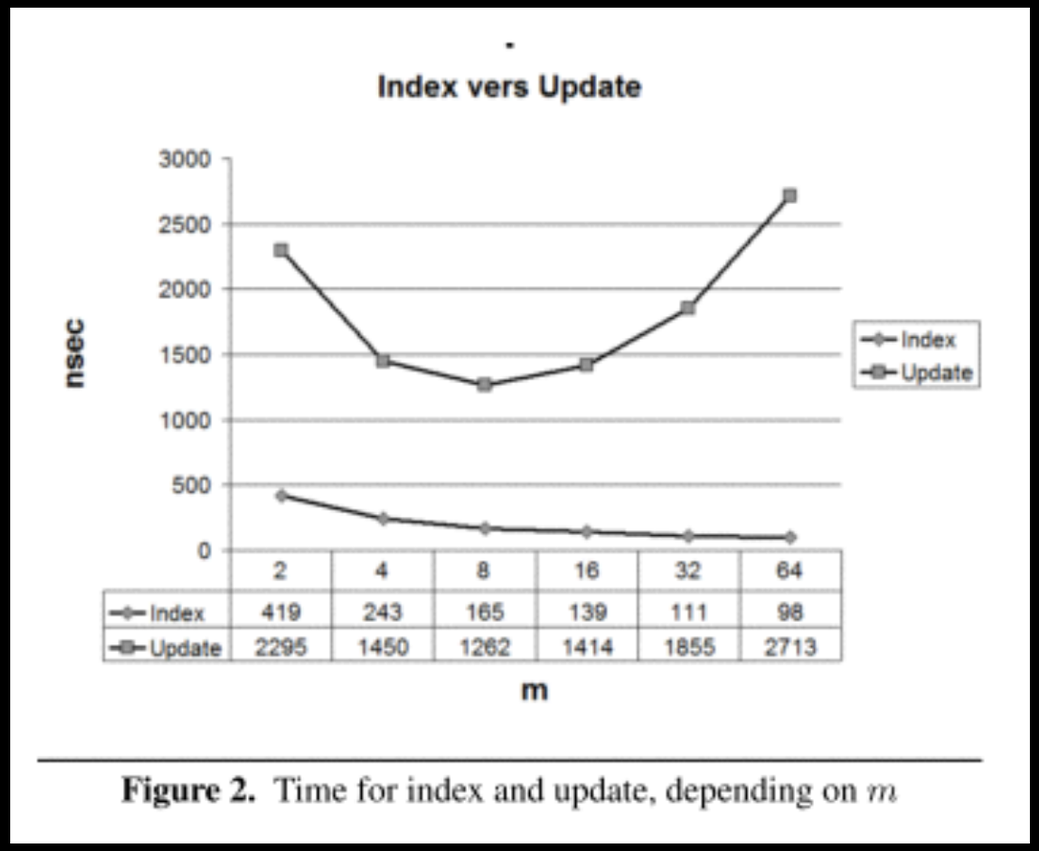


Figure 2. Time for index and update, depending on m

From Bagwell, Rompf 2011

$$32^7$$

34,359,738,368 elements

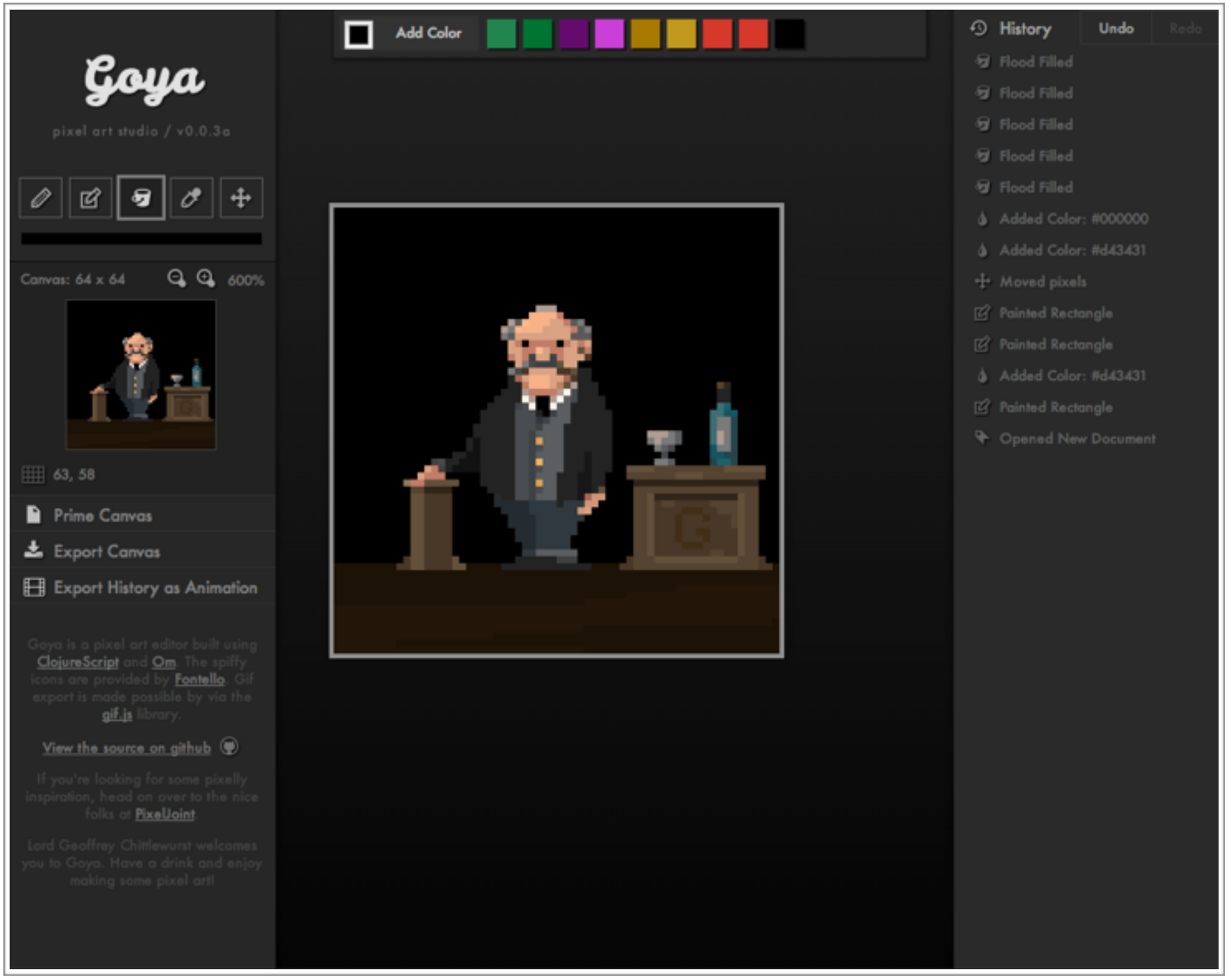
demo


React | A JavaScript library
facebook.github.io/react/
React
React
A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

$$f(D_0) = V_0$$

$$f(D_1) = V_1$$

$$\mathrm{diff}(\mathrm{V}_0,\mathrm{V}_1) = \mathrm{CHANGES}$$


Goya
pixel art studio / v0.0.3a
Add Color
History
Undo
Redo
Flood Filled
Flood Filled
Flood Filled
Flood Filled
Flood Filled
Added Color: #000000
Added Color: #d43431
Moved pixels
Painted Rectangle
Painted Rectangle
Added Color: #d43431
Painted Rectangle
Opened New Document
Canvas: 64 x 64
600%
63, 58
Prime Canvas
Export Canvas
Export History as Animation
Goya is a pixel art editor built using ClojureScript and Om. The spiffy icons are provided by Fontello. Gif export is made possible by via the gif.js library.
View the source on github
If you're looking for some pixelly inspiration, head on over to the nice folks at PixelJoint.
Lord Geoffrey Chittlewurst welcomes you to Goya. Have a drink and enjoy making some pixel art!

demo

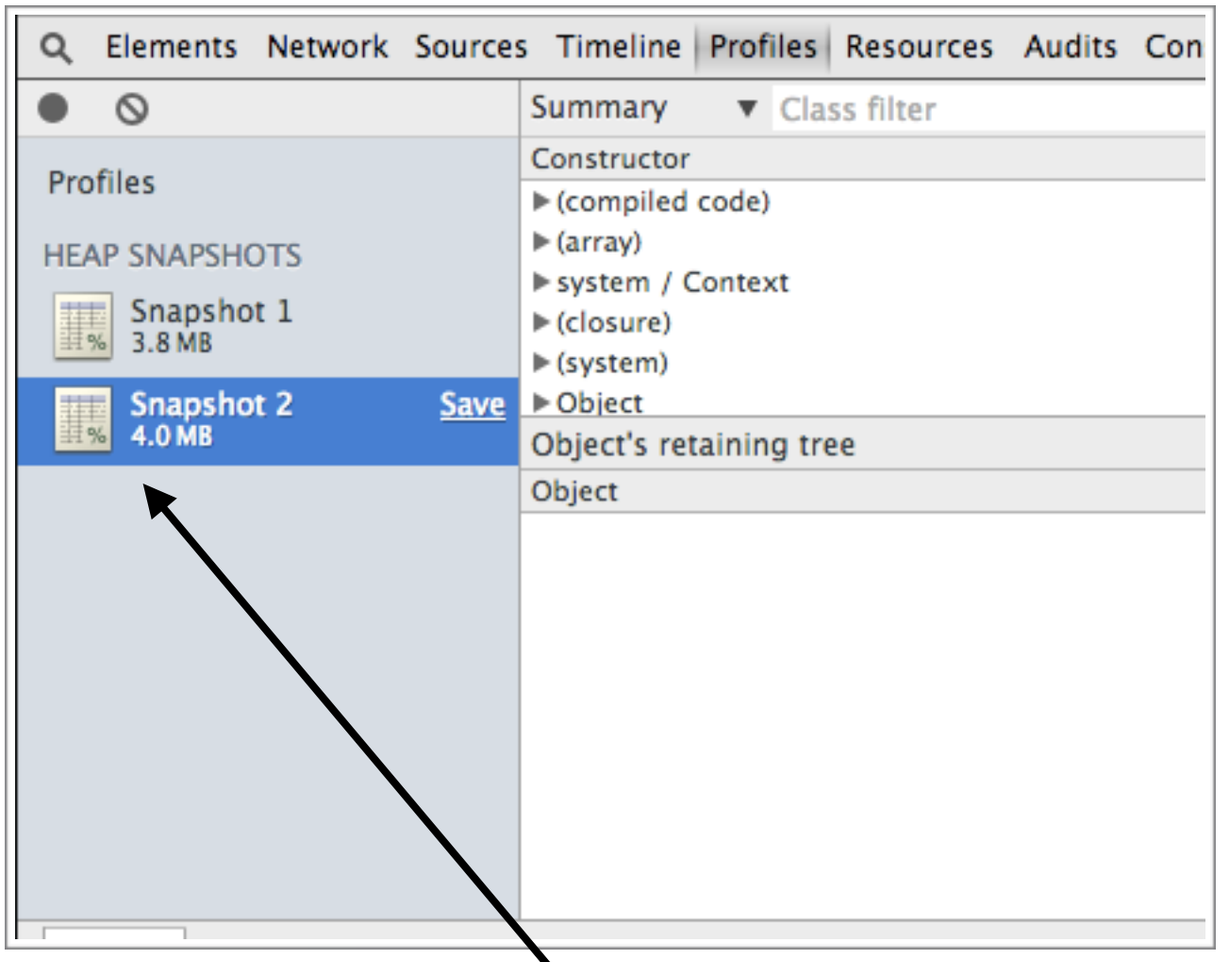
branch: master · goya / src / cljs / goya / timemachine.cljs

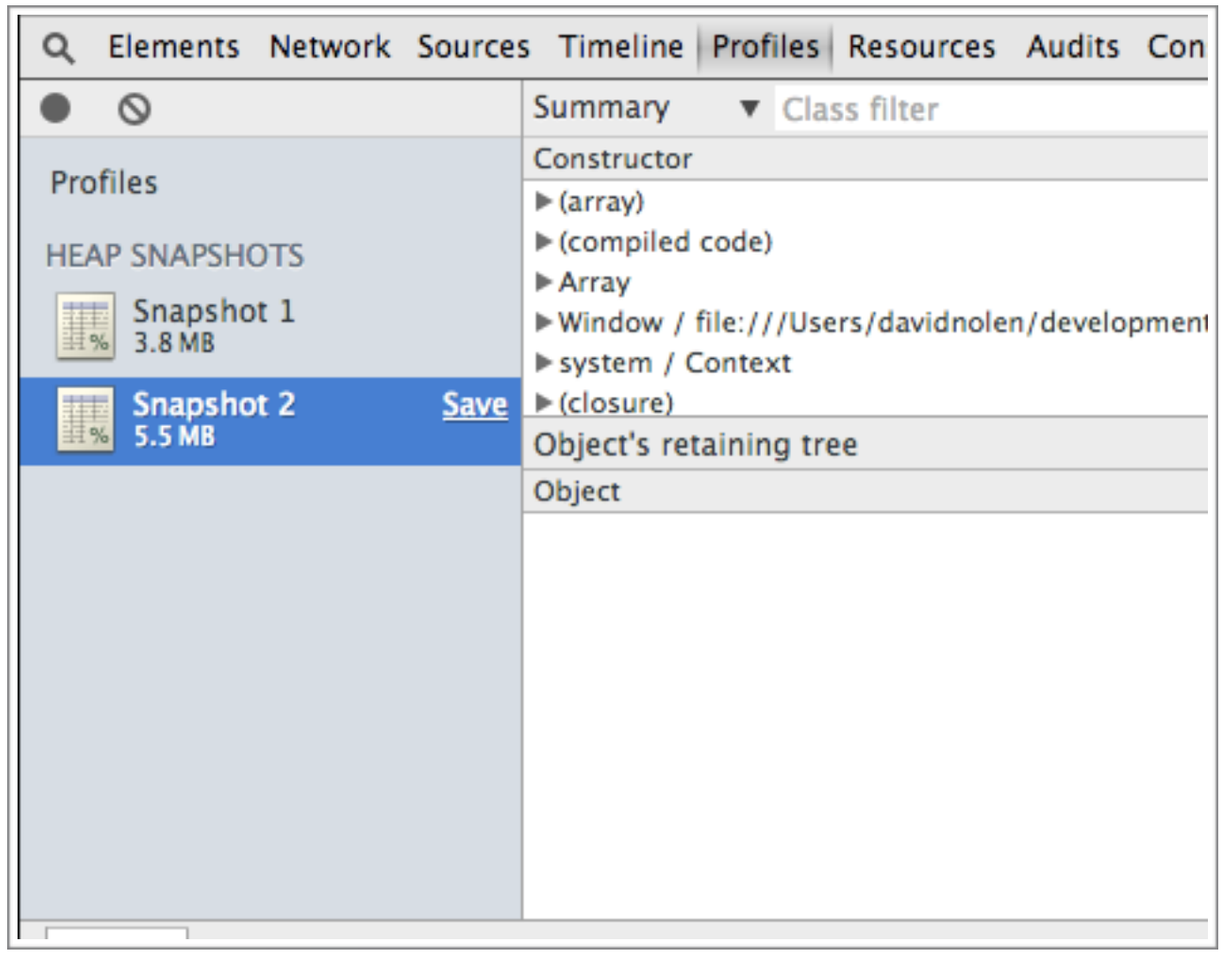
swannodette 13 days ago Project layout refactor, better production settings

1 contributor

file 62 lines (41 sloc) 1.85 kb — Open | Edit | Raw | Blame | History | Delete

```
(ns goya.timemachine
        (:require [goya.appstate :as app]
            [goya.previewstate :as previewstate]))


;; ================================================================================
;; Credits to David Nolen's Time Travel blog post.

(def app-history (atom [(get-in @app/app-state [:main-app])]))
(def app-future (atom []))



;; ================================================================================

(defn update-preview []
  (reset! previewstate/preview-state
          (assoc-in @previewstate/preview-state [:main-app :image-data]
                    (get-in @app/app-state [:main-app :image-data]))))

(defn show-history-preview [idx]
  (reset! previewstate/preview-state
          (assoc-in @previewstate/preview-state [:main-app :image-data]
                    (get-in (nth @app-history idx) [:image-data]))))

(add-watch app/app-state :preview-watcher
  (fn [_ _ _ _] (update-preview)))


(defn undo-is-possible []
  (> (count @app-history) 1))

(defn redo-is-possible []
  (> (count @app-future) 0))


(defn push-onto-undo-stack [new-state]
  (let [old-watchable-app-state (last @app-history)]
    (when-not (= old-watchable-app-state new-state)
      (swap! app-history conj new-state))))


(defn do-undo []
  (when (undo-is-possible)
    (swap! app-future conj (last @app-history))
    (swap! app-history pop)
    (reset! app/app-state (assoc-in @app/app-state [:main-app] (last @app-history)))))

(defn do-redo []
  (when (redo-is-possible)
    (reset! app/app-state (assoc-in @app/app-state [:main-app] (last @app-future)))
    (push-onto-undo-stack (last @app-future))
    (swap! app-future pop)))


(defn handle-transaction [tx-data root-cursor]
  (when (= (:tag tx-data) :add-to-undo)
    (reset! app-future [])
    (let [new-state (get-in (:new-state tx-data) [:main-app])]
      (push-onto-undo-stack new-state))))
```





Persistent Data Structures ... ROCK

facebook / **immutable-js**

Unwatch 96 | Unstar 2,683 | Fork 85

Immutable Data Collections for Javascript

368 commits | **1** branch | **14** releases | **15** contributors

branch: **master** | **immutable-js** / +

Update README.md ···

leebyron authored 9 days ago — latest commit `9fedc9883a`

__tests__	Ensure equality works correctly for Set. #96	21 days ago
dist	Ensure equality works correctly for Set. #96	21 days ago
resources	lowercase require module name, simplifying case-sensitive file systems.	a month ago
src	Ensure equality works correctly for Set. #96	21 days ago
type-definitions	renamed deepMerge -> mergeDeep	14 days ago
.gitignore	Clean up gruntfile, add dist	3 months ago
CONTRIBUTING.md	Moving over to fb's team github page	2 months ago
Gruntfile.js	Use unminified source in node, minified in scripts. #69	a month ago

Code

Issues 27

Pull Requests 0

Wiki

Pulse

Graphs

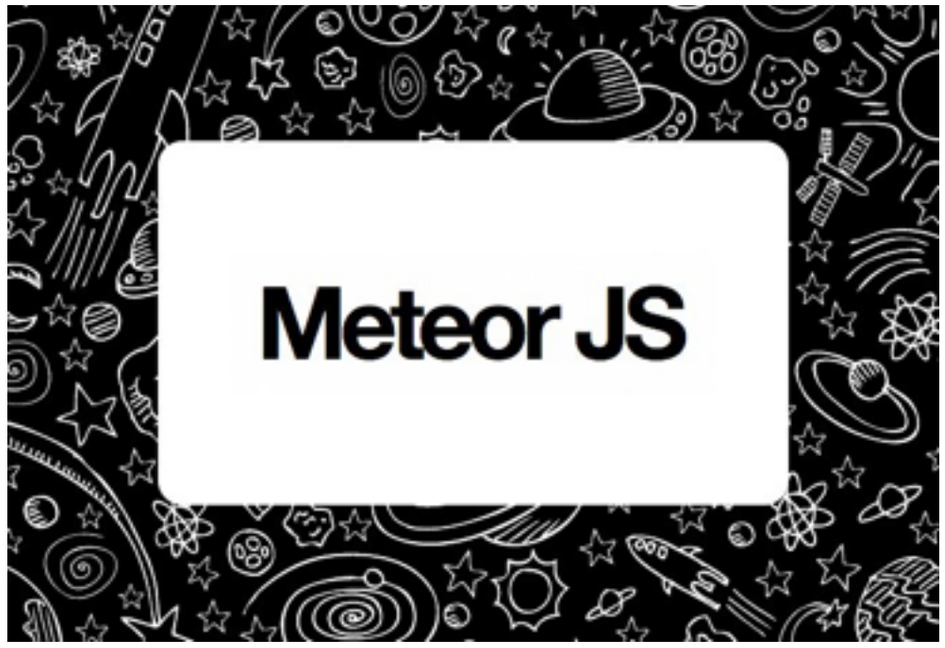
HTTPS clone URL

https://github.com

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP


Meteor JS


circleci


Prismatic

CircleCI demo

Interactivity

- Whole program optimization
- Code splitting via Google Closure Modules
- Extensible REPL architecture, run interactively wherever JavaScript runs!

Questions?