

# A surprising real world 70x performance jump



PRAGMA  
CONFERENCE



# Sparkle



# Sparkle

A screenshot of a web page editor interface, likely Wix, showing a travel website for "Balloon Travel". The main canvas displays a vibrant image of several hot air balloons floating over a landscape of rolling hills and mountains at sunset. Overlaid on this image is a large, stylized title "Balloon Travel" in yellow and white. Below the main image, there's a section titled "Ballooning Sites" featuring three smaller images of balloons and their locations: Serengeti, Tanzania; Albuquerque, New Mexico; and Queenstown, New Zealand. A footer section contains the text "To most people, the sky is the limit. To those" followed by a house icon and the text "Page 1". The top navigation bar includes options like Zoom, Device, Grid, Page, Blog, Text, Layout Block, Box, Image, Add, SEO, Settings, Preview, and Publish. On the right side, there's a detailed sidebar for managing the "Image" element, showing settings for filename ("pasted-image.png"), title (tooltip), and description. It also includes options for replacing or removing the image, adjusting its size (1x, 2x, 3x), and applying various styles like border, rounding, shadow, opacity, and filters. A note indicates that the image resolution is optimal for 3x, though it can be made larger if desired. Below these settings, there are sections for "Edit Background", "On Click", and "Scroll Effects".



# Sparkle

Text Wrap 65% Zoom Device Grid Page Blog Text Layout Block Box Image Add SEO Settings Preview Publish Style Arrange Page

Bitmap (PNG, JPEG, GIF)

FILENAME hot-air-balloon-4761\_1920.png

TITLE (TOOLTIP)

DESCRIPTION

1920x1440 px Replace... Remove

Stretch  Fill  Fit

1x 2x 3x

Image resolution sufficient for 2x, larger image or smaller element required for 3x.

Edit Background  
Note: background removal is optimized for foreground subjects with a good contrast over the background.

Border  
 Rounding  
 Shadow  
 Opacity  
 Filters

**On Click**  
Do Nothing

**Scroll Effects**  
None

Pages Layers

Text Wrap

Page 1

The image shows a hot air balloon against a blue sky. The balloon is multi-colored with vertical stripes of red, orange, yellow, green, blue, and purple. The basket is dark and contains two people. The text wrap feature is applied to the image, with text floating around it. The text is a placeholder: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tincidunt lorem enim, eget fringilla turpis congue vitae. Phasellus aliquam nisi ut lorem vestibulum eleifend. Nulla ut arcu non nisi congue venenatis vitae ut ante. Nam iaculis sem nec ultrices dapibus. Phasellus eu ultrices turpis. Vivamus non mollis lacus, non ullamcorper nisl. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus sit amet scelerisque ipsum. Morbi nulla dolor, adipiscing non sagittis, vitae dignissim justo sollicitudin. Phasellus non varius lacus, aliquet feugiat mauris. Phasellus fringilla commodo sem vel pellentesque. Ut porttitor tincidunt risus a pharetra. Cras nec vestibulum massa. Mauris sagittis leo a libero convallis accumsan. Aenean ut mollis ipsum. Donec aliquam egestas convallis. Fusce dapibus, neque sed mattis consectetur, erat nibh vulputate sapien, ac accumsan arcu sem quis nibh. Etiam et mi sed mauris commodo tristique. Proin mollis elementum purus, a porta quam vehicula et. Quisque ullamcorper, sapien ut egestas faucibus, tortor mauris tempor odio, sed pretium risus dui sit amet lectus. Sed ligula mi, tincidunt nec porttitor vel, aliquet sit amet libero. Nulla sagittis ultricies sem, non pretium augue bibendum vitae. Nunc luctus tristique urna eu tincidunt. Etiam ultricies neque ante, ut placerat dolor dapibus. Curabitur luctus orci et gravida laoreet. Sed ultrices id nulla id mollis. Donec tempor dapibus sem, a convallis felis elementum sed. Aenean nisi tortor, dictum ac massa non, rhoncus sagittis leo. Interdum et malesuada fames ac ante ipsum primis in



# Sparkle

The image shows a user interface for a page editor, likely a drag-and-drop website builder. At the top, there's a toolbar with various icons for zooming, device simulation, grid, page, blog, text, layout block, box, image, and add. The main area has tabs for Pages and Layers. A central content area contains a large image of a colorful hot air balloon against a blue sky. Text is wrapped around the balloon, appearing as if it's floating within the basket. The text is a standard placeholder text (Lorem ipsum). On the left side, there's a sidebar with a preview of the page, showing the hot air balloon and some text snippets. On the right side, there's a panel for managing images, showing the file name 'hot-air-balloon-4761\_1920.png' and options for replace, remove, stretch, fill, and fit. There are also checkboxes for edit background, border, rounding, shadow, opacity, and filters, as well as sections for on click and scroll effects.



# Sparkle

Balloon Travel

65% Zoom Device Grid Page Blog Text Layout Block Box Image Add SEO Settings Preview Publish

Style Arrange Page

**Image**

Bitmap (PNG, JPEG, GIF)

FILENAME  
pasted-image.png

TITLE (TOOLTIP)

DESCRIPTION

1834x2362 px Replace... Remove

Stretch  Fill  Fit

1x 2x 3x +

Image resolution is optimal for 3x, element can be made larger if desired.

Edit Background  
Note: background removal is optimized for foreground subjects with a good contrast over the background.

Border

Rounding

Shadow

Opacity

Filters

**On Click**

Do Nothing

**Scroll Effects**

Animate While Scrolling

Pages Layers

Balloon Travel

Home Sites Packages Book

Balloon Travel

Serengati, Tanzania

Albuquerque, New Mexico

Gisborne, New Zealand

To most people, the sky is the limit. To those

Page 1

Ballooning

**Sites**

Balloon Travel



Sparkle



balloon-1.png



balloon-2.png



fiji.jpg



# Sparkle



balloon-1.png



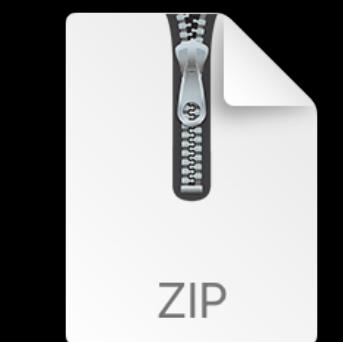
balloon-2.png



fuji.jpg



paper.pdf



download.zip



# Sparkle



balloon-1.png



balloon-2.png

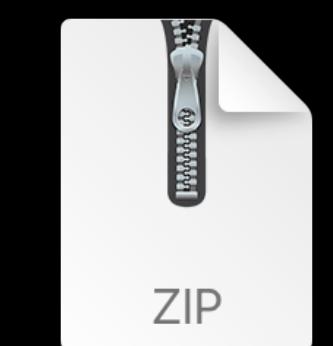
Website  
model



fuji.jpg



paper.pdf



download.zip



Sparkle



website.sparkle



# Sparkle



# Sparkle

- Objective-C runtime hacks
- Low level filesystem stuff
- Obscure SQL

...for fun & profit!

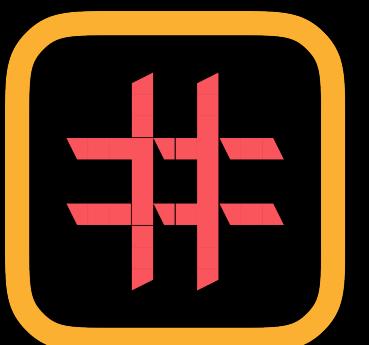
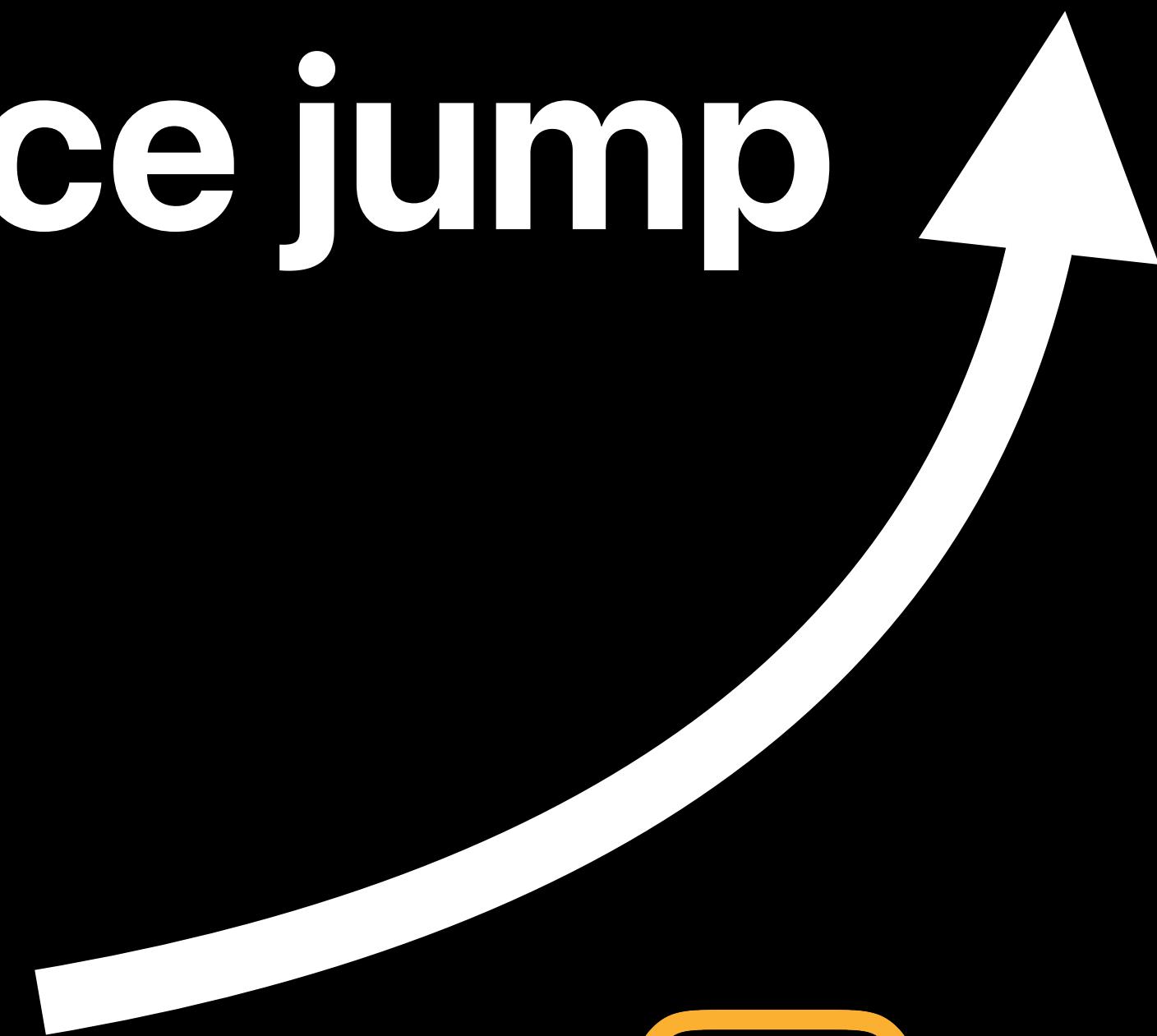
# A surprising real world 70x performance jump



PRAGMA  
CONFERENCE

# A surprising real world 70x performance jump

clickbait



PRAGMA  
CONFERENCE

# A surprising real world 70x performance jump

(unintentional)  
clickbait



PRAGMA  
CONFERENCE

A surprising real world ~30x  
performance jump

(unintentional)  
clickbait



PRAGMA  
CONFERENCE

# The problem



website.sparkle



website.sparkle



website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



## large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



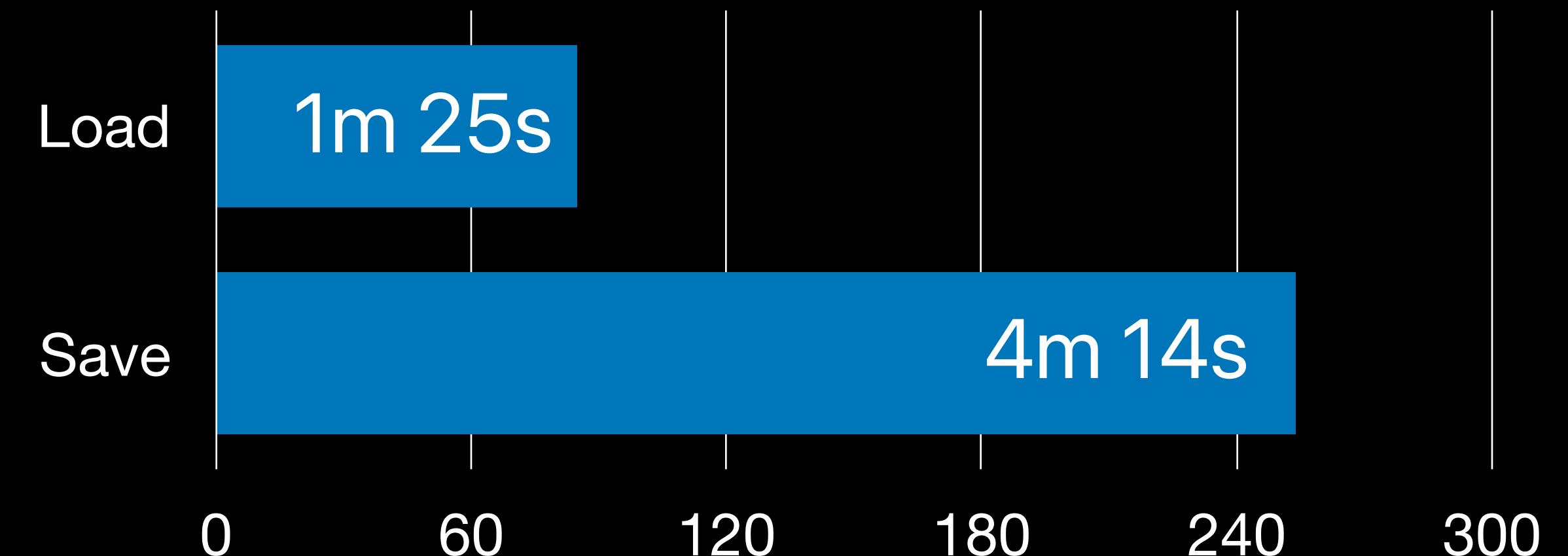
large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



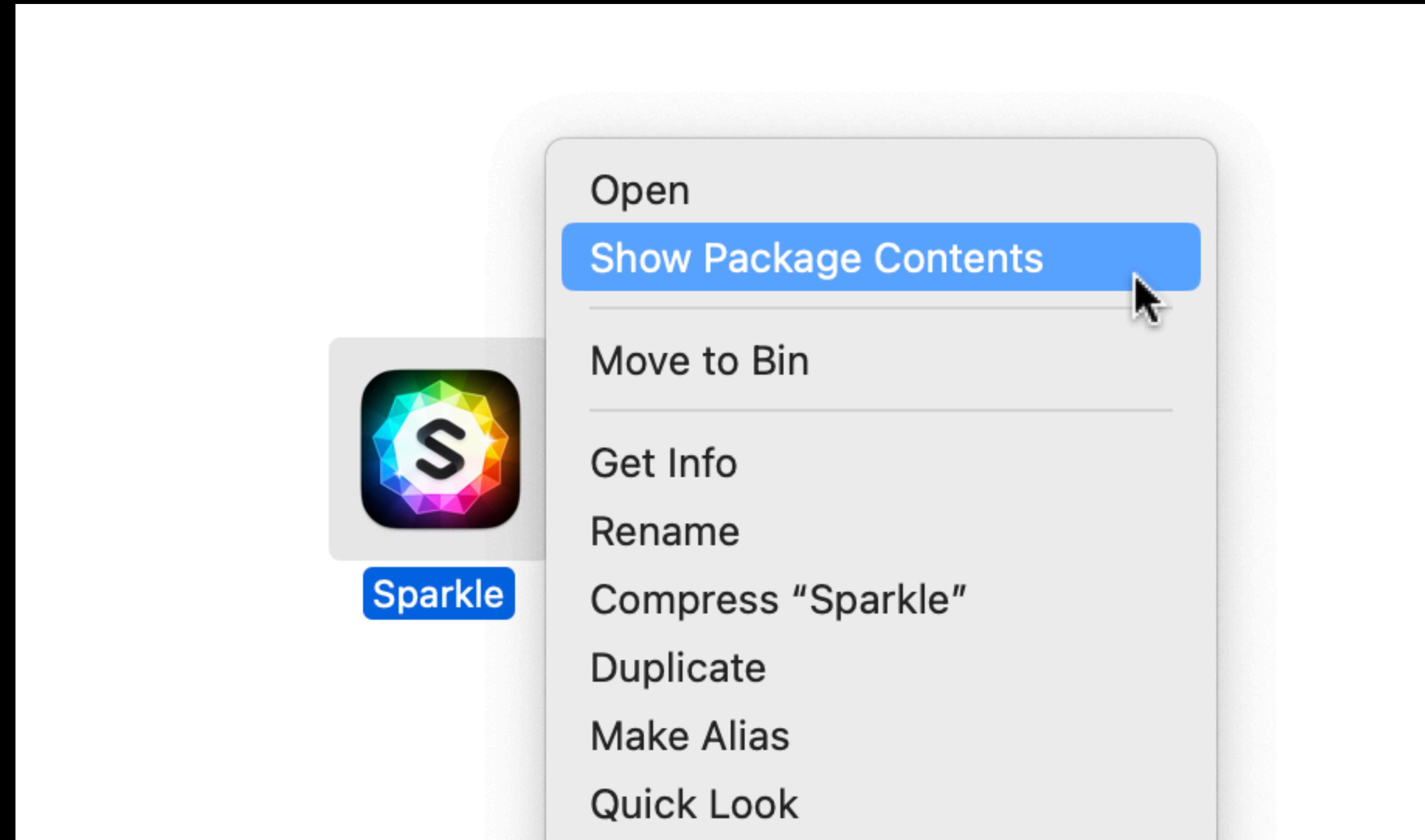
## large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



Why not use a file bundle?

# File bundle



# File bundle issues

- Hard to share
- Backup issues
- Cloud sync issues

# File bundle issues

- Hard to share
- Backup issues
- Cloud sync issues

Procreate Dreams:

“All your assets are bundled into one file made for iCloud synchronization and backups”

```
% unzip -v test.pages
Archive: test.pages
      Length   Method    Size   Cmpr      Date    Time    CRC-32     Name
----- -----
        4749  Stored     4749   0% 03-06-2023 18:19  3431e399  Index/Document.iwa
         543  Stored      543   0% 03-06-2023 18:19  fe3e0e70  Index/ViewState.iwa
         296  Stored      296   0% 03-06-2023 18:19  b93f544b  Index/CalculationEngine-5037.iwa
          22  Stored       22   0% 07-03-2023 15:44  22c83e08  Index/AnnotationAuthorStorage-5036.iwa
      40230  Stored    40230   0% 03-06-2023 18:19  ef39583f  Index/DocumentStylesheet-5040.iwa
         23  Stored       23   0% 07-03-2023 15:44  22979e87  Index/DocumentMetadata.iwa
     11573  Stored    11573   0% 03-06-2023 18:19  19f46a4b  Index/Metadata.iwa
         322  Stored      322   0% 03-06-2023 18:19  4ab82032  Metadata/Properties.plist
          36  Stored       36   0% 03-06-2023 18:19  99ccdf24  Metadata/DocumentIdentifier
         273  Stored      273   0% 03-06-2023 18:19  4e89ec53  Metadata/BuildVersionHistory.plist
     13351  Stored    13351   0% 03-06-2023 18:19  2e1610ee  preview.jpg
         829  Stored      829   0% 03-06-2023 18:19  3a4909c1  preview-micro.jpg
        1425  Stored     1425   0% 03-06-2023 18:19  b4fd7f37  preview-web.jpg
-----
      73672           73672   0%
```

13 files

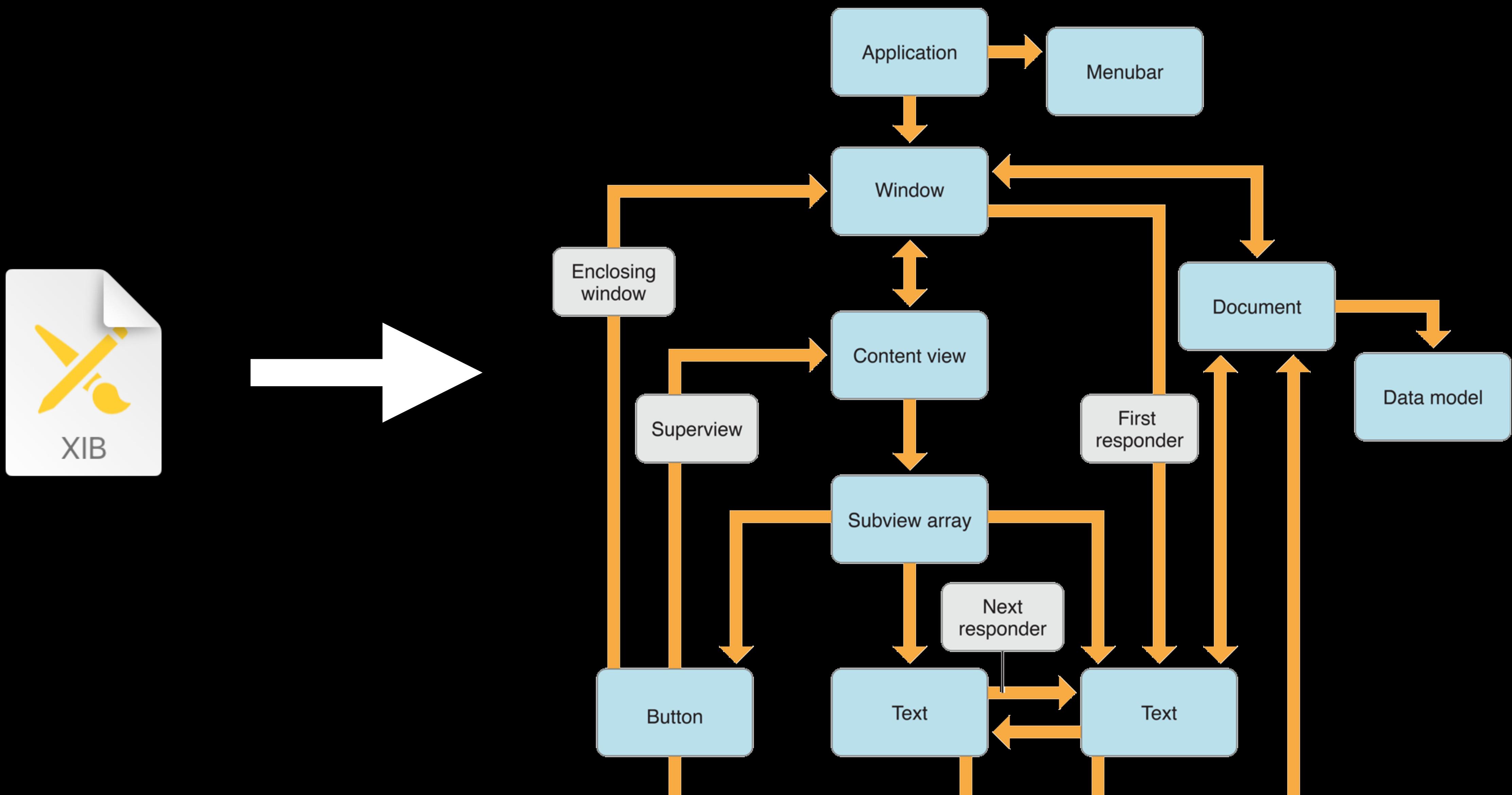
# Why not use Core Data?

# Improving load time

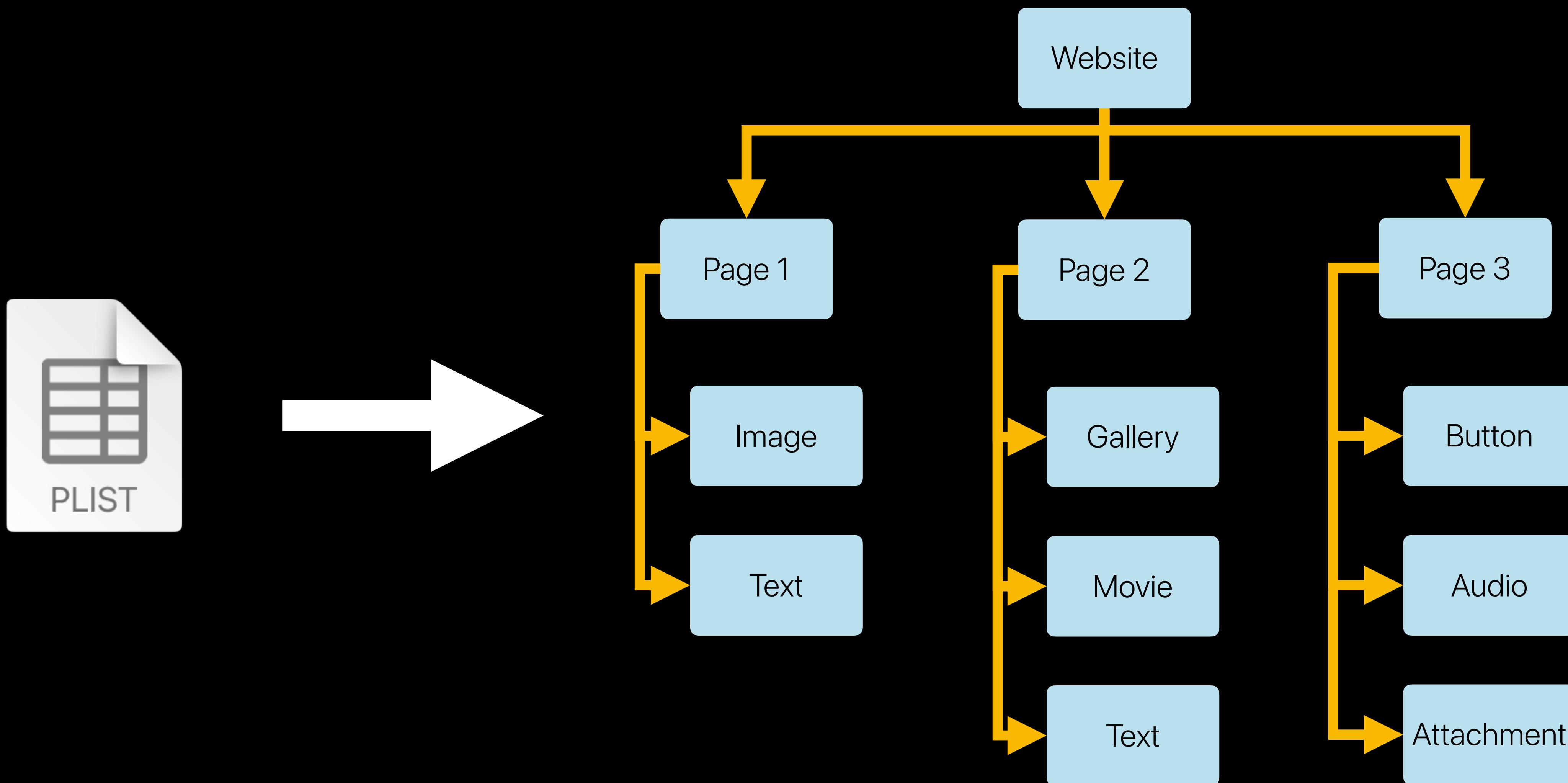
# Improving load time

- Don't do unnecessary work

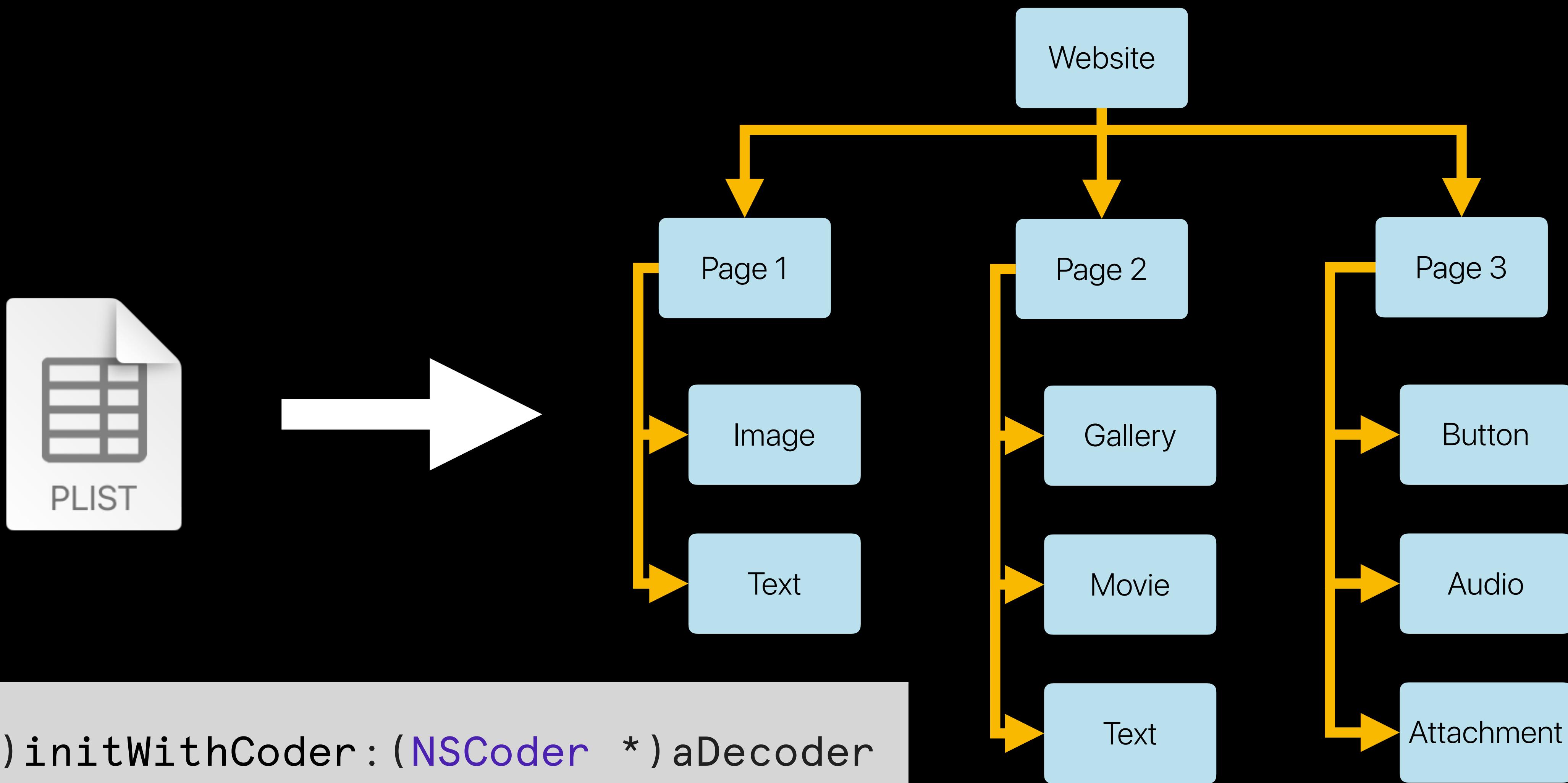
# Improving load time



# Improving load time



# Improving load time



# Improving load time

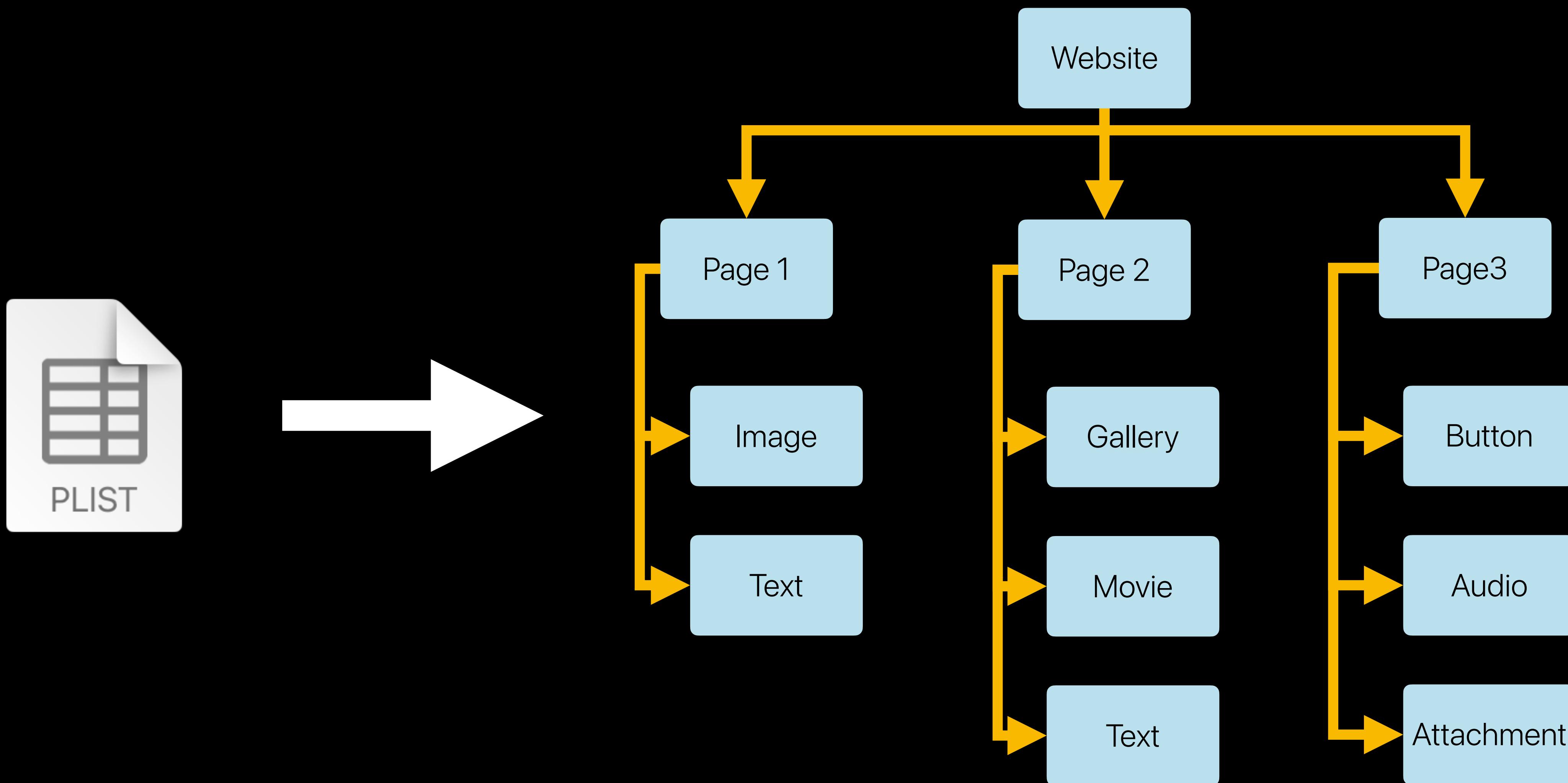
# Improving load time

- Don't do unnecessary work

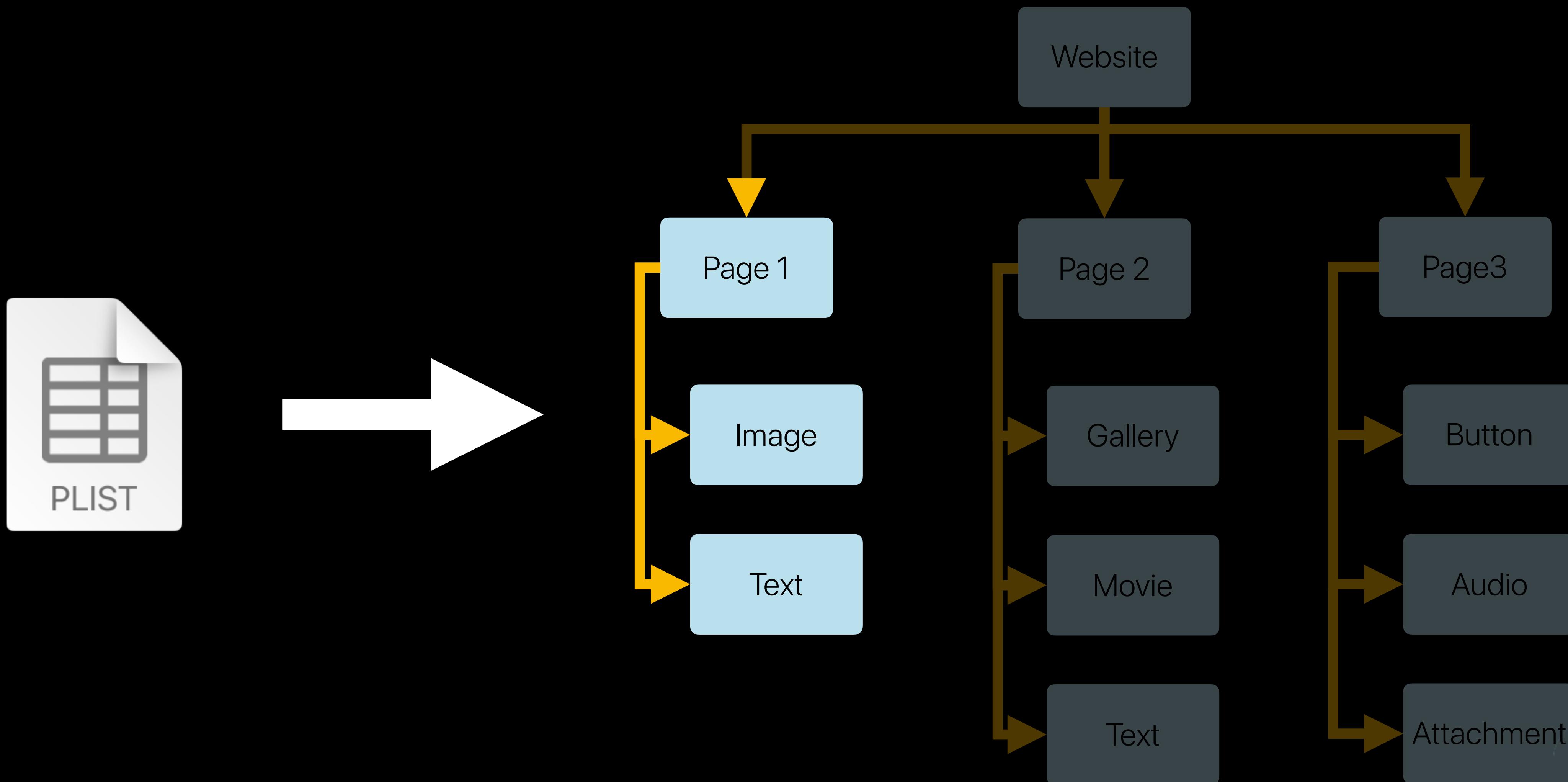
# Improving load time

- Don't do unnecessary work
- Only load objects that are used

# Improving load time



# Improving load time



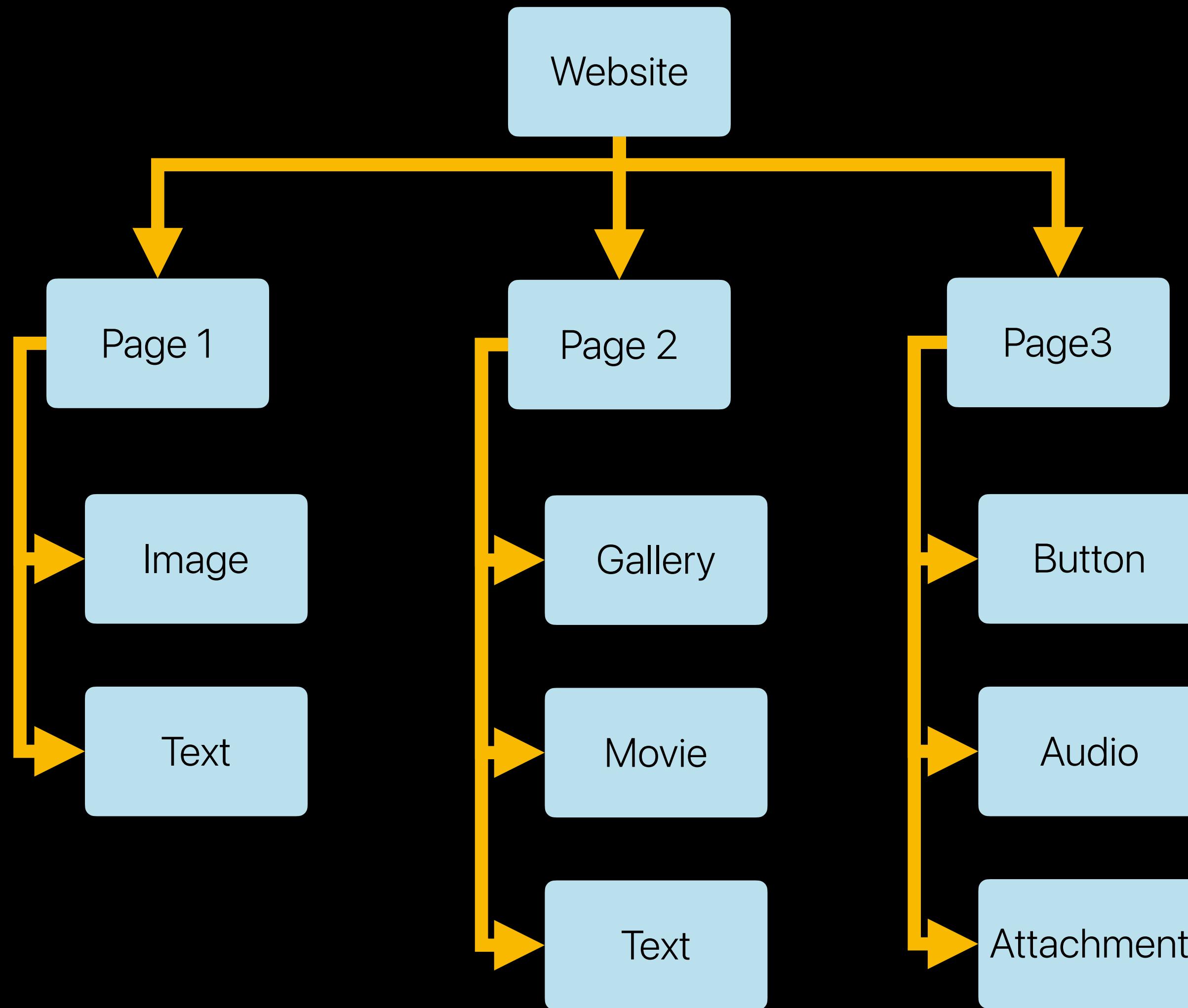
# Improving load time

- Don't do unnecessary work
- Only load objects that are used

# Improving load time

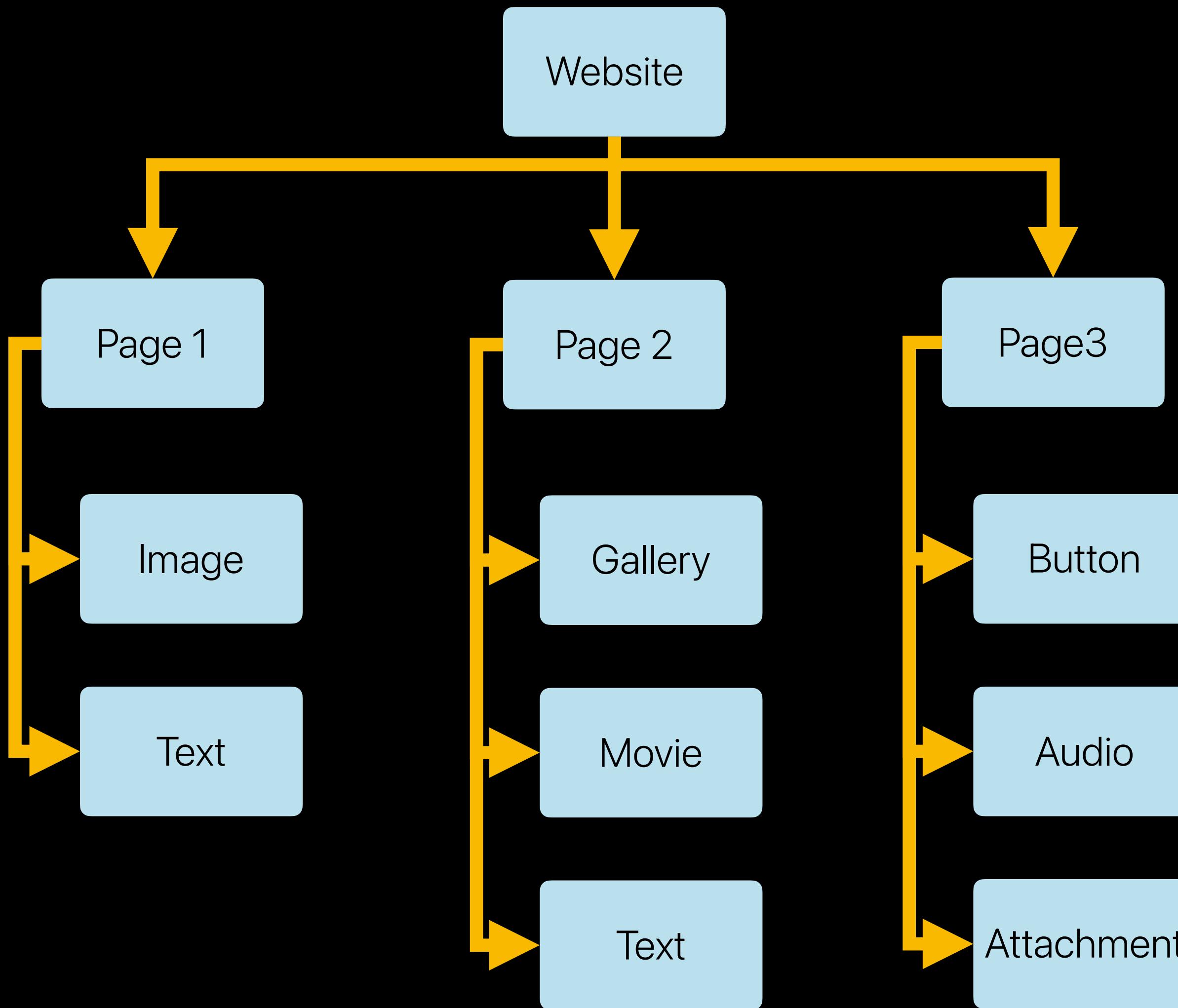
- Don't do unnecessary work
- Only load objects that are used
- Lazy load objects, on demand

# Improving load time

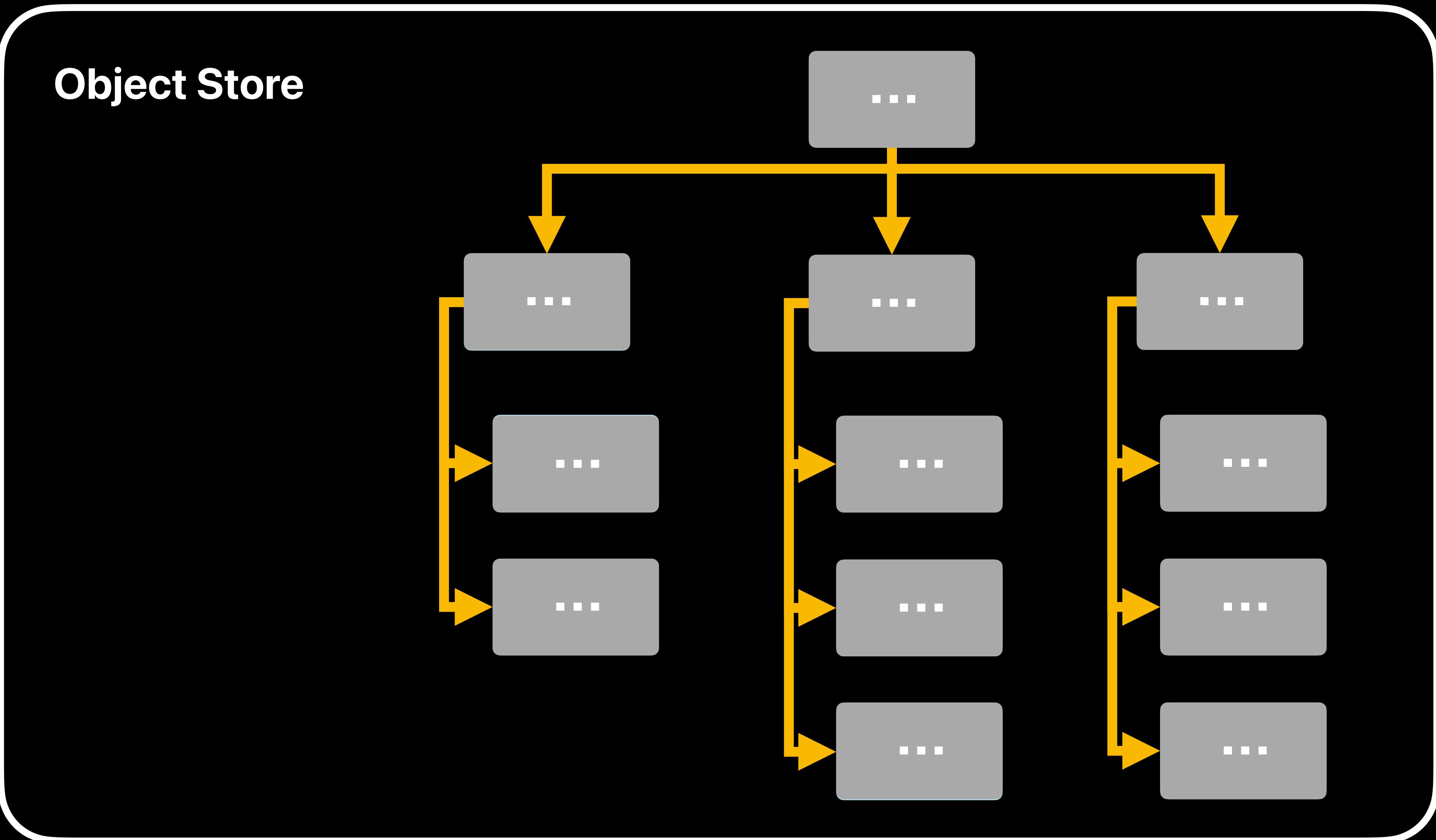


# Improving load time

## Object Store



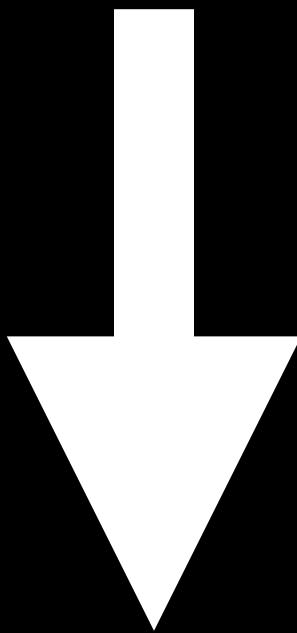
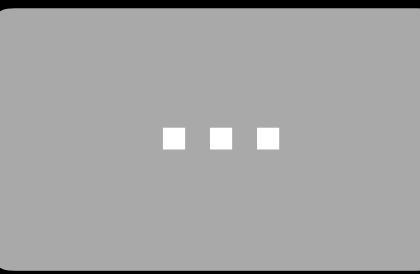
# Improving load time



# Improving load time

```
// Apple:  
// NSProxy is an abstract superclass defining an API  
// for objects that act as stand-ins for other objects  
// or for objects that don't exist yet.  
  
@interface LazyLoaderProxy : NSProxy  
...  
@end
```

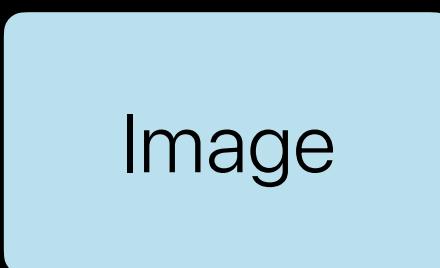
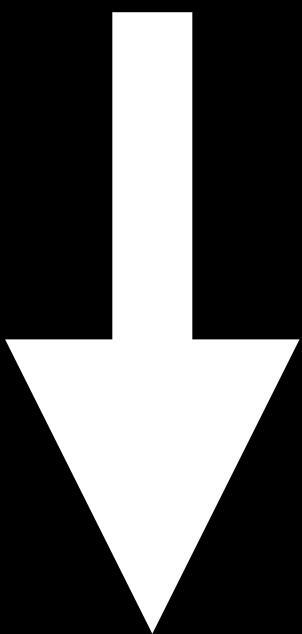
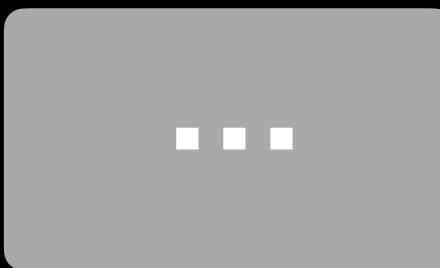
# Improving load time



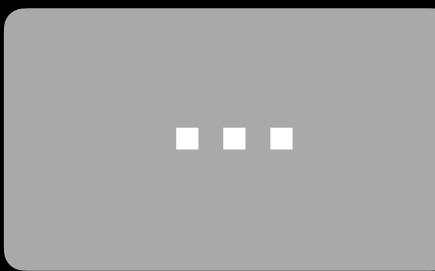
???

Image

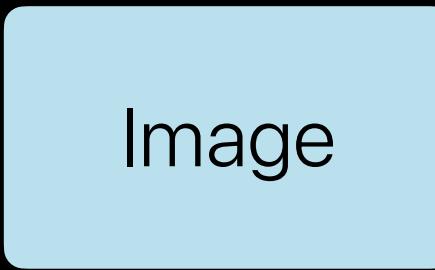
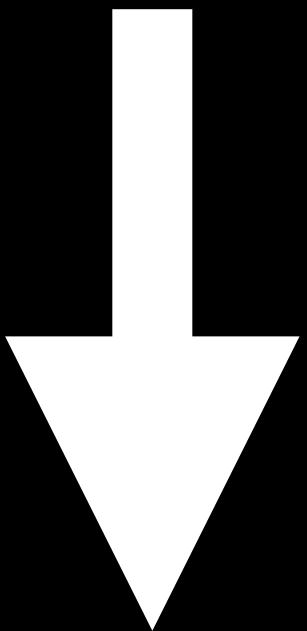
# Improving load time



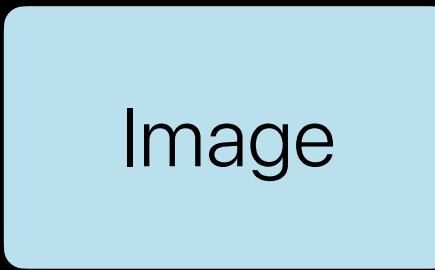
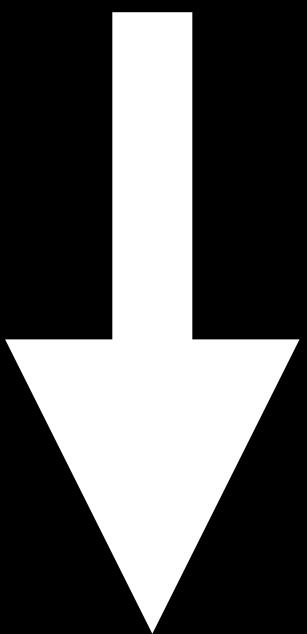
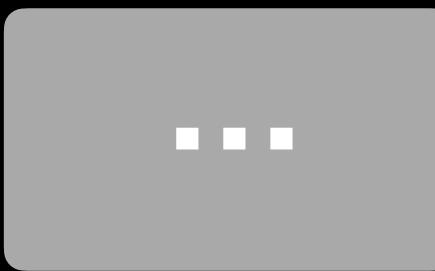
# Improving load time



- Instantiate NSProxy

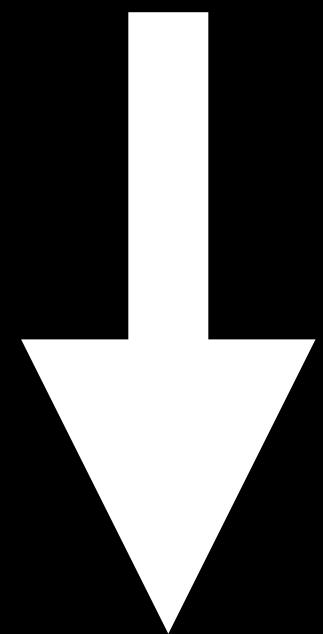
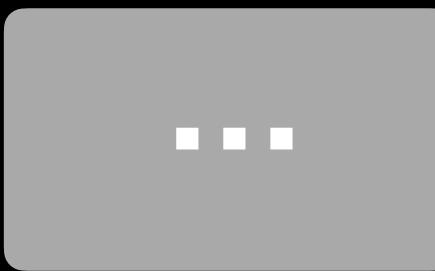


# Improving load time



- Instantiate NSProxy
- Detect use of object

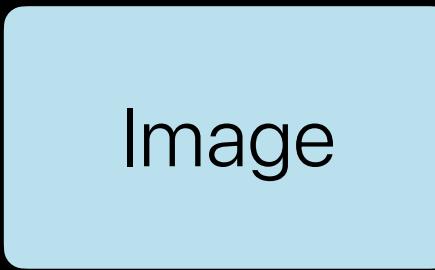
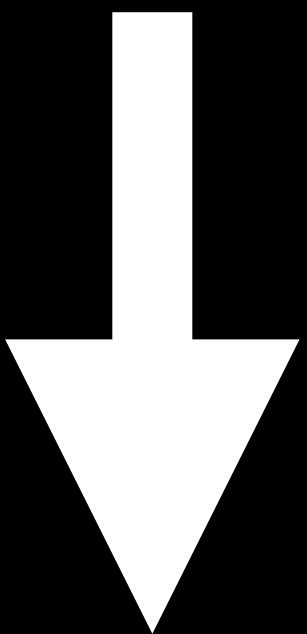
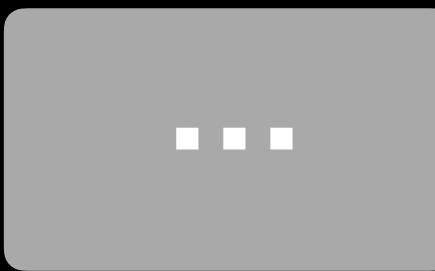
# Improving load time



Image

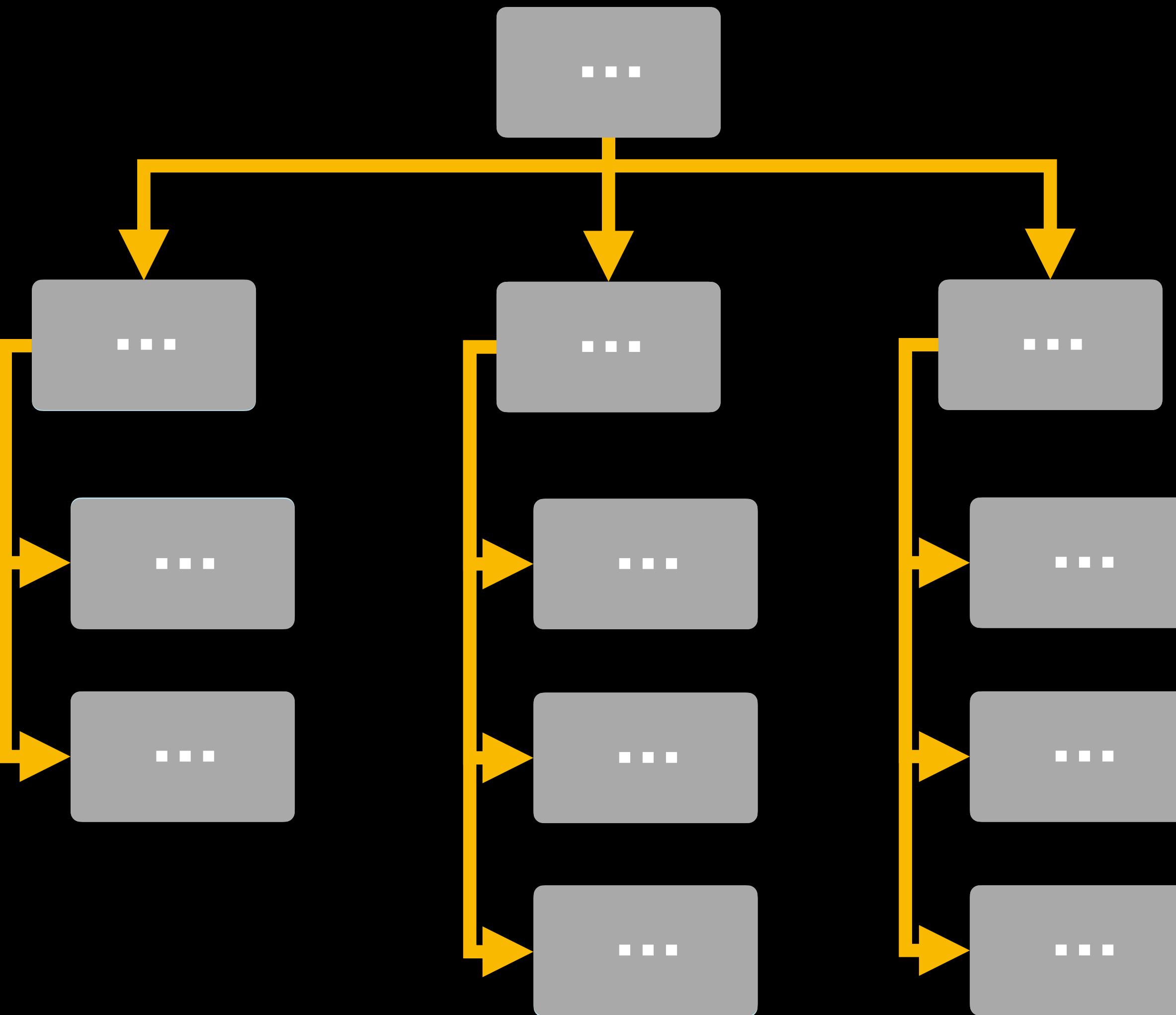
- Instantiate NSProxy
- Detect use of object
- Transform proxy instance into object

# Improving load time

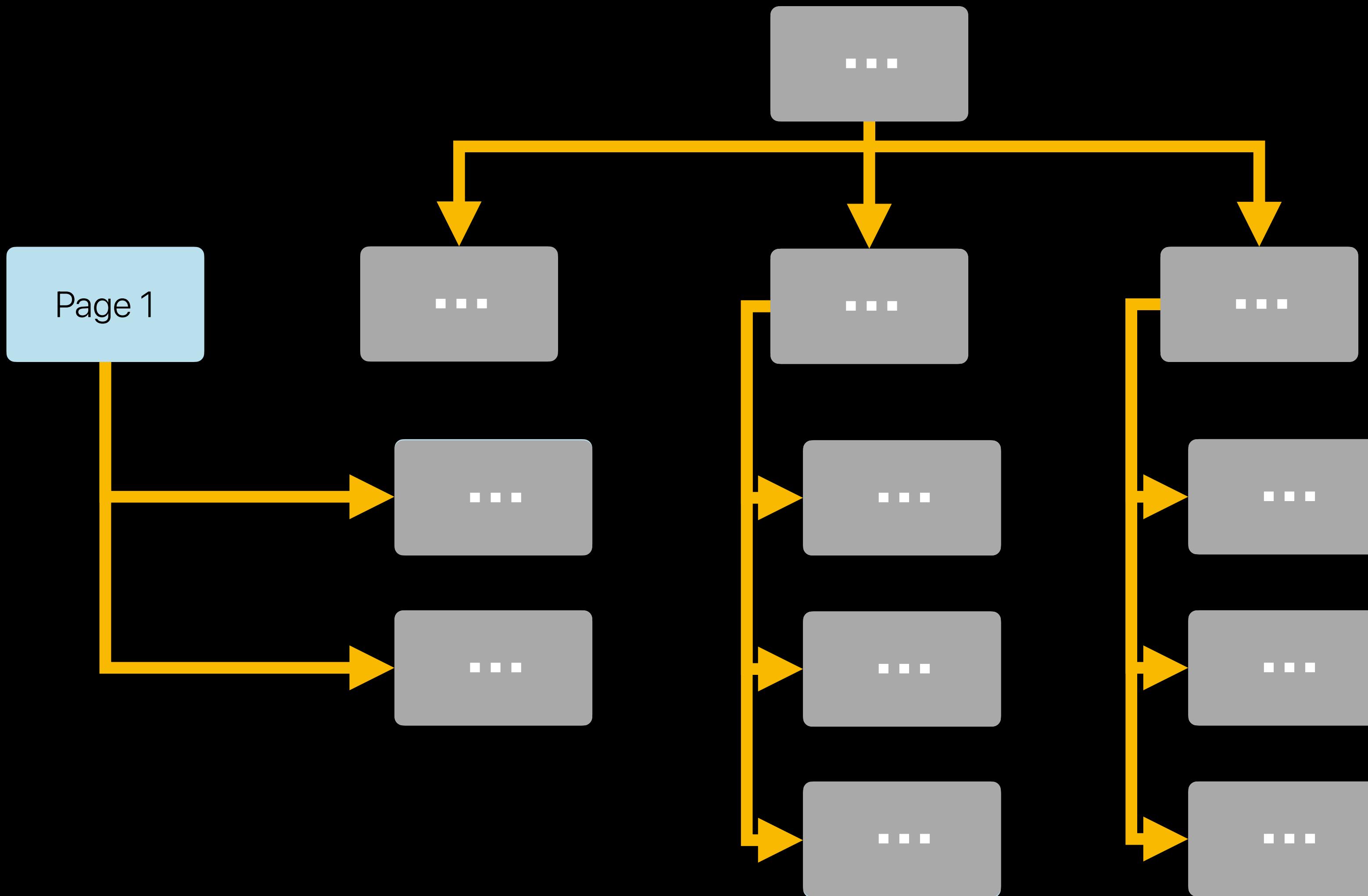


- Instantiate NSProxy
- Detect use of object
- Transform proxy instance into object
- Dearchive object

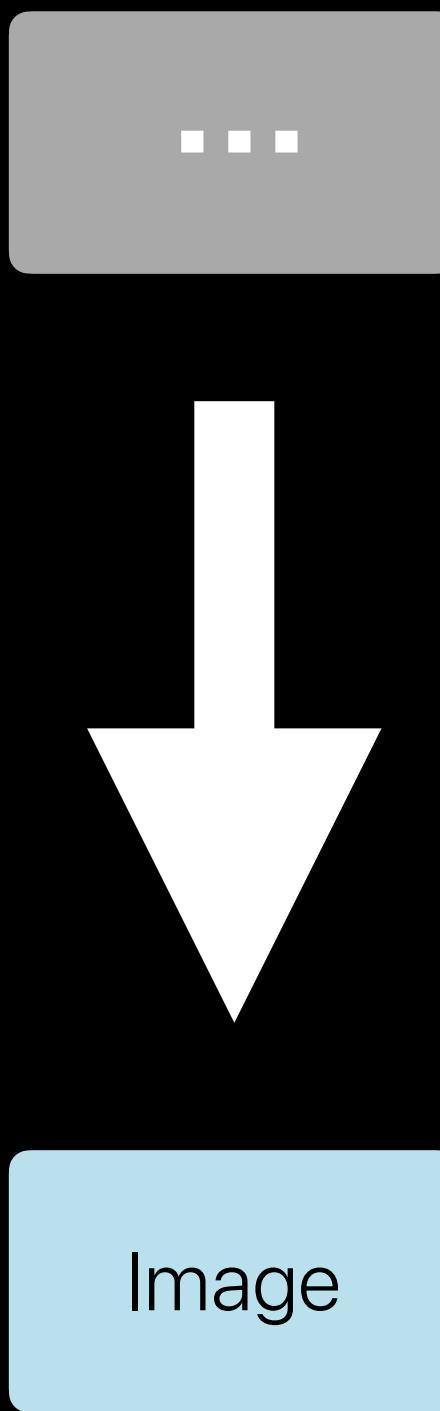
# Improving load time



# Improving load time



# Improving load time



- Instantiate NSProxy
- Detect use of object
- Transform proxy instance into object
- Dearchive object

# Improving load time

## Memory layout

...

Image

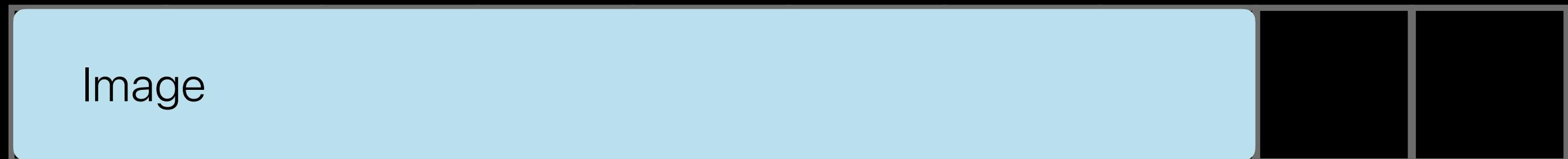
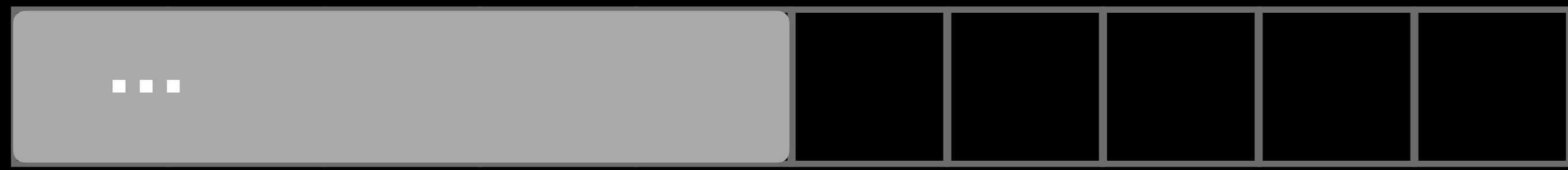
# Improving load time

## Memory layout

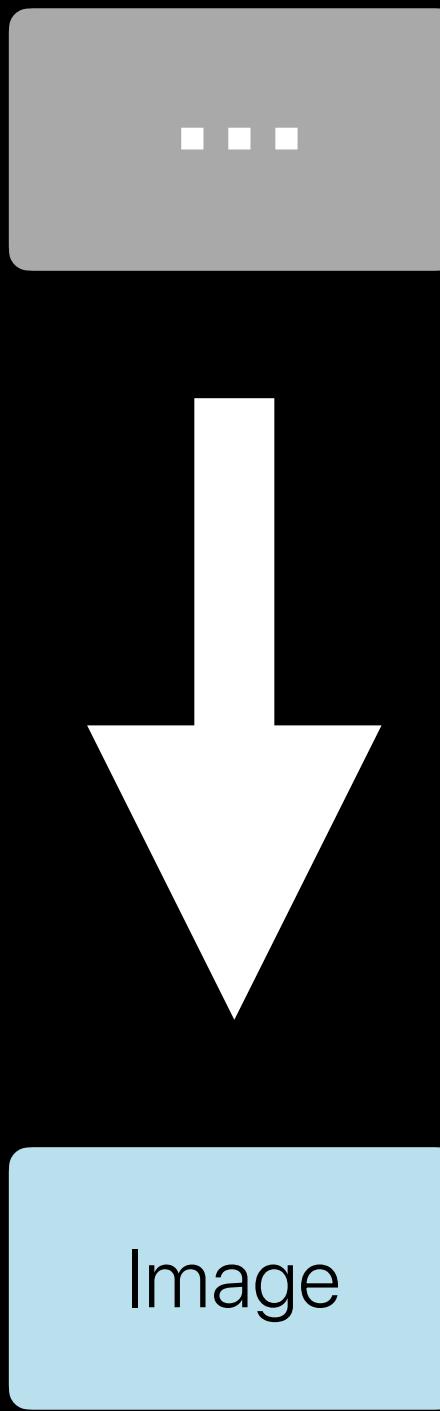


# Improving load time

## Memory layout

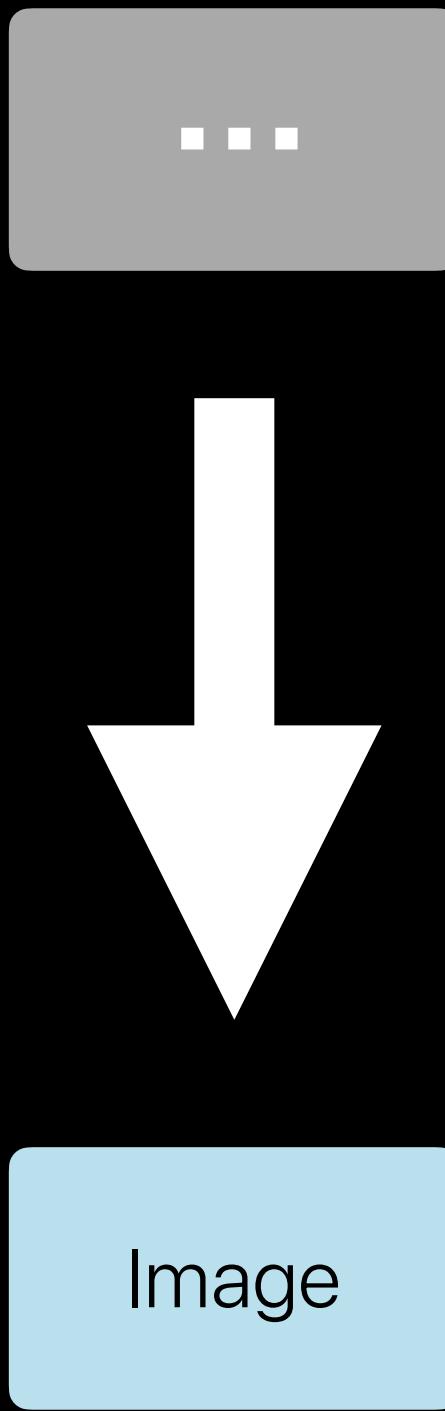


# Improving load time



- Instantiate NSProxy
- Detect use of object
- Transform proxy instance into object
- Dearchive object

# Improving load time



- Measure object sizes
- Instantiate NSProxy with custom size
- Detect use of object
- Transform proxy instance into object
- Dearchive object

# Improving load time

```
NSInteger size1 = class_getInstanceSize(targetObjectClass);
NSInteger size2 = class_getInstanceSize(LazyLoaderProxy.class);
NSInteger allocsize = MAX(size1, size2);

// allocate object memory
void *bytes = calloc(1, allocsize);

// non-ARC code
id obj = objc_constructInstance(LazyLoaderProxy.class, bytes);
```

# Improving load time

```
NSInteger size1 = class_getInstanceSize(targetObjectClass);
NSInteger size2 = class_getInstanceSize(LazyLoaderProxy.class);
NSInteger allocsize = MAX(size1, size2);

// allocate object memory
void *bytes = calloc(1, allocsize);

// non-ARC code
id obj = objc_constructInstance(LazyLoaderProxy.class, bytes);
```

# Improving load time

```
NSInteger size1 = class_getInstanceSize(targetObjectClass);
NSInteger size2 = class_getInstanceSize(LazyLoaderProxy.class);
NSInteger allocsize = MAX(size1, size2);

// allocate object memory
void *bytes = calloc(1, allocsize);

// non-ARC code
id obj = objc_constructInstance(LazyLoaderProxy.class, bytes);
```

# Improving load time

```
NSInteger size1 = class_getInstanceSize(targetObjectClass);
NSInteger size2 = class_getInstanceSize(LazyLoaderProxy.class);
NSInteger allocsize = MAX(size1, size2);

// allocate object memory
void *bytes = calloc(1, allocsize);

// non-ARC code
id obj = objc_constructInstance(LazyLoaderProxy.class, bytes);
```

# Improving load time

```
- (NSMethodSignature *)methodSignatureForSelector:(SEL)sel
{
    Method m = class_getInstanceMethod(targetObjectClass, sel);
    return [NSMethodSignature
        signatureWithObjCTypes:method_getTypeEncoding(m)];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    // do something
}
```

# Improving load time

```
- (NSMethodSignature *)methodSignatureForSelector:(SEL)sel
{
    Method m = class_getInstanceMethod(targetObjectClass, sel);
    return [NSMethodSignature
        signatureWithObjCTypes:method_getTypeEncoding(m)];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    // do something
}
```

# Improving load time

```
- (NSMethodSignature *)methodSignatureForSelector:(SEL)sel
{
    Method m = class_getInstanceMethod(targetObjectClass, sel);
    return [NSMethodSignature
        signatureWithObjCTypes:method_getTypeEncoding(m)];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    // do something
}
```

# Improving load time

```
+ (BOOL)loadObject:(LazyLoaderProxy *)proxy
{
    memset((void *)proxy, 0, allocsize);
    object_setClass(proxy, targetObjectClass); // non-ARC code
    [our_object_store instantiateIntoObject:proxy];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    [LazyLoaderProxy loadObject:self];
    [invocation invokeWithTarget:self];
}
```

# Improving load time

```
+ (BOOL)loadObject:(LazyLoaderProxy *)proxy
{
    memset((void *)proxy, 0, allocsize);
    object_setClass(proxy, targetObjectClass); // non-ARC code
    [our_object_store instantiateIntoObject:proxy];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    [LazyLoaderProxy loadObject:self];
    [invocation invokeWithTarget:self];
}
```

# Improving load time

```
+ (BOOL)loadObject:(LazyLoaderProxy *)proxy
{
    memset((void *)proxy, 0, allocsize);
    object_setClass(proxy, targetObjectClass); // non-ARC code
    [our_object_store instantiateIntoObject:proxy];
}

- (void)forwardInvocation:(NSInvocation *)invocation
{
    [LazyLoaderProxy loadObject:self];
    [invocation invokeWithTarget:self];
}
```

# Improving load time



large website.sparkle

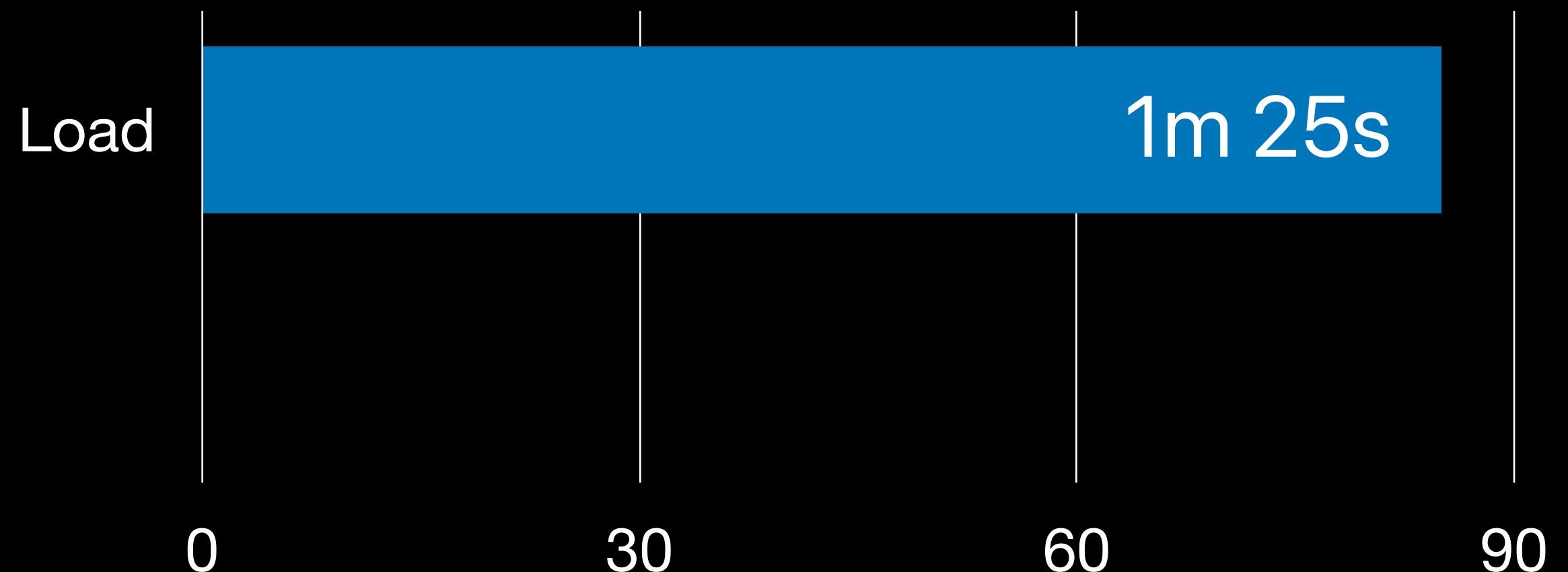
- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

# Improving load time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

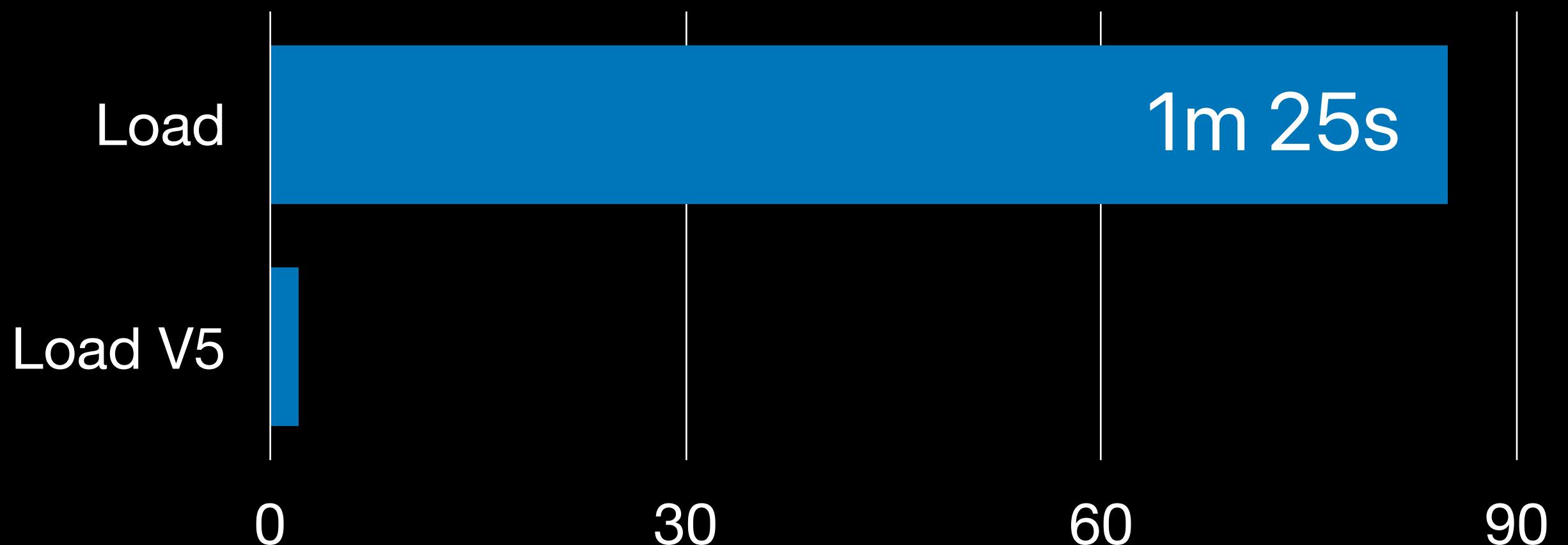


# Improving load time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

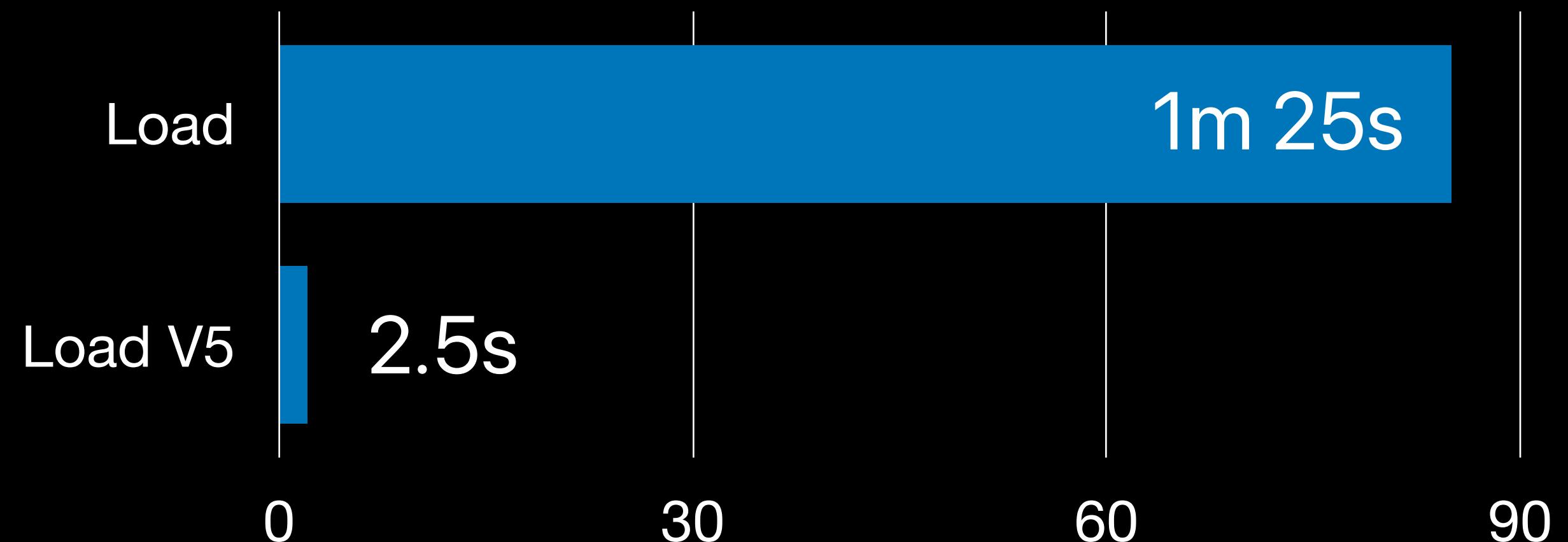


# Improving load time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



# Improving load time

# Improving load time

Pitfalls:

# Improving load time

## Pitfalls:

- Reinventing the wheel

# Improving load time

## Pitfalls:

- Reinventing the wheel
- Custom dearchive slower

# Improving load time

## Pitfalls:

- Reinventing the wheel
- Custom dearchive slower
- Lazy load != never load

# Improving load time

## Pitfalls:

- Reinventing the wheel
- Custom dearchive slower
- Lazy load != never load
- Trouble with post-load side effects

# Improving load time

## Pitfalls:

- Reinventing the wheel
- Custom dearchive slower
- Lazy load != never load
- Trouble with post-load side effects
- Not loading = not using memory

# Objective-C

# Objective-C

- Lazy loading not possible without Objective-C's dynamic runtime

# Objective-C

- Lazy loading not possible without Objective-C's dynamic runtime
- Swift? We would have looked for a different solution

# Objective-C

- Lazy loading not possible without Objective-C's dynamic runtime
- Swift? We would have looked for a different solution
- There never was a retain or a release in Sparkle

# Improving save time

# Improving save time

- Don't do unnecessary work

# Improving save time

```
% time cp "large website.sparkle" copy.sparkle
cp "large website.sparkle" copy.sparkle 0.02s user 14.83s system
19% cpu 1:17.87 total
```

# Improving save time

```
% time cp "large website.sparkle" copy.sparkle
cp "large website.sparkle" copy.sparkle 0.02s user 14.83s system
19% cpu 1:17.87 total
```

# Improving save time

```
% time cp -c "large website.sparkle" clone.sparkle  
cp -c "large website.sparkle" clone.sparkle 0.00s user 0.00s  
system 57% cpu 0.007 total
```

# Improving save time

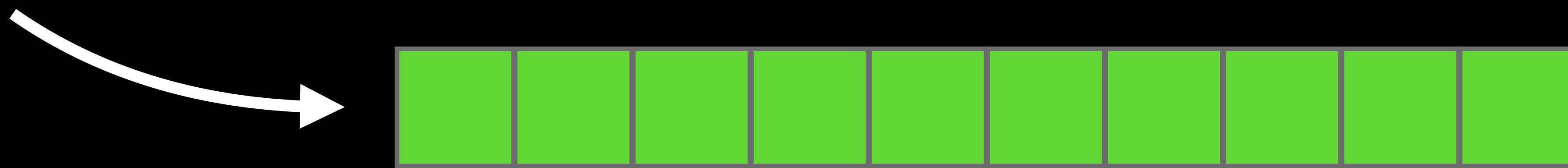
```
% time cp -c "large website.sparkle" clone.sparkle  
cp -c "large website.sparkle" clone.sparkle 0.00s user 0.00s  
system 57% cpu 0.007 total
```

# Improving save time

cpu 0.007 total

# Copy

large website.sparkle

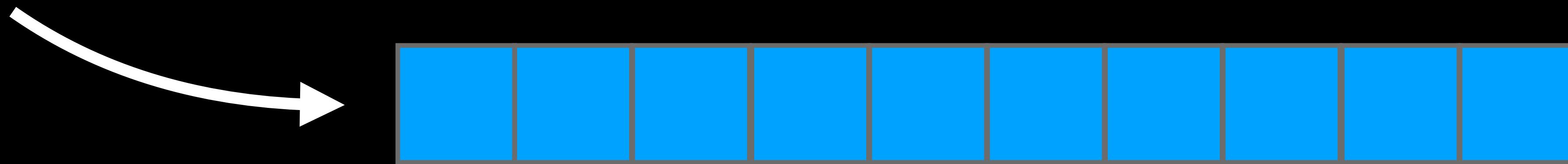


# Copy

large website.sparkle

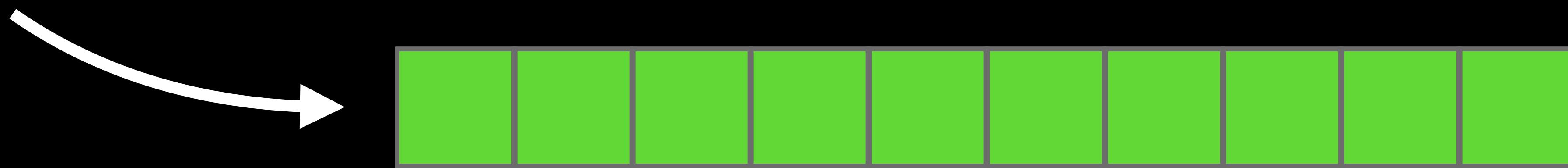


copy.sparkle



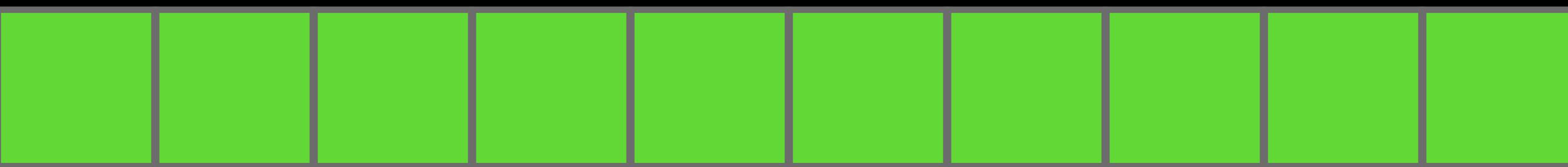
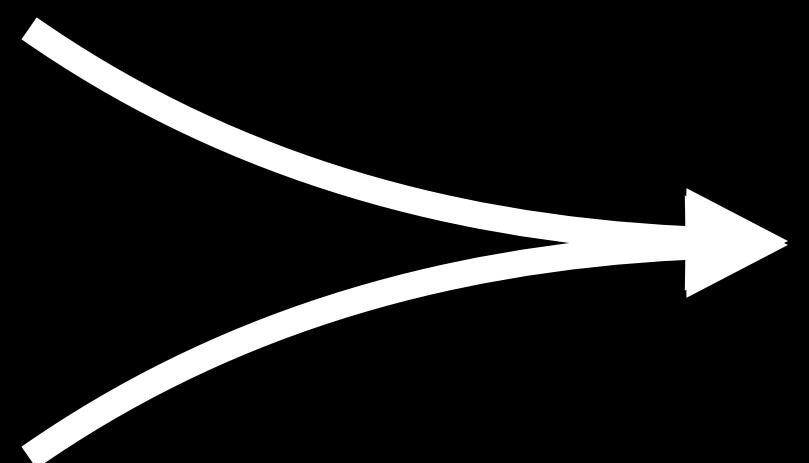
# Copy on Write

large website.sparkle



# Copy on Write

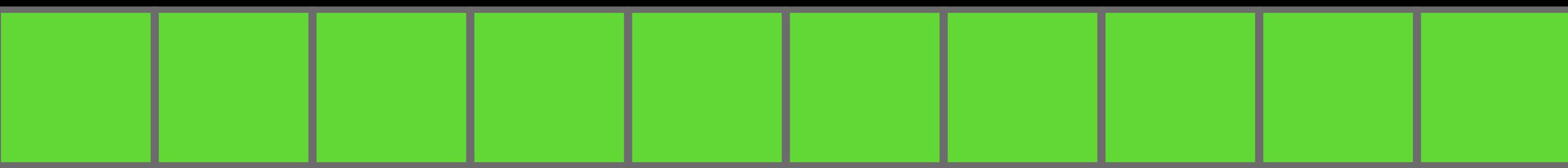
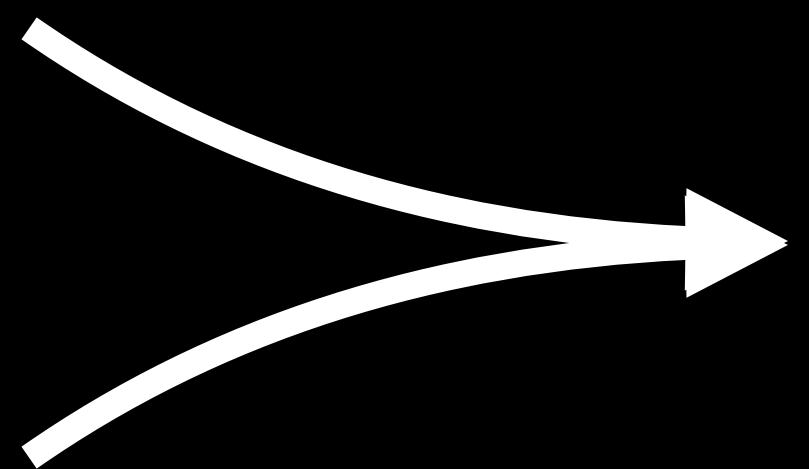
large website.sparkle



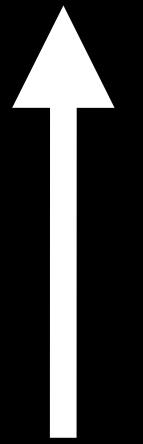
clone.sparkle

# Copy on Write

large website.sparkle

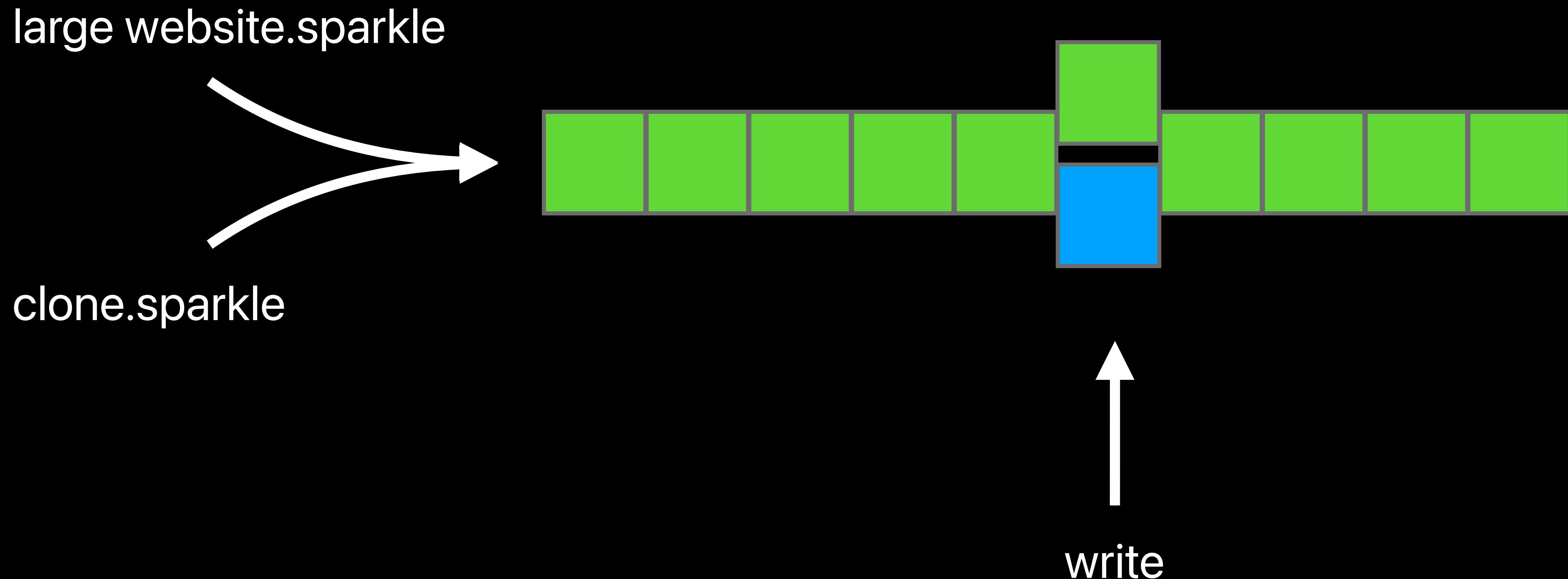


clone.sparkle



write

# Copy on Write



# Improving save time

# Improving save time



# Improving save time



# Improving save time



# Improving save time



# Improving save time



SQLITE

# Improving save time



SQLITE

# Improving save time



[website.sparkle](http://website.sparkle)

# Improving save time



[website.sparkle](http://website.sparkle)

# Improving save time

# Improving save time

Plan:

# Improving save time

Plan:

- Clone file

# Improving save time

Plan:

- Clone file
- Go through all objects

# Improving save time

Plan:

- Clone file
- Go through all objects
  - Archive + save object

# Improving save time

Plan:

- Clone file
- Go through all objects
  - Archive + save object
- Remove old objects from the clone

# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

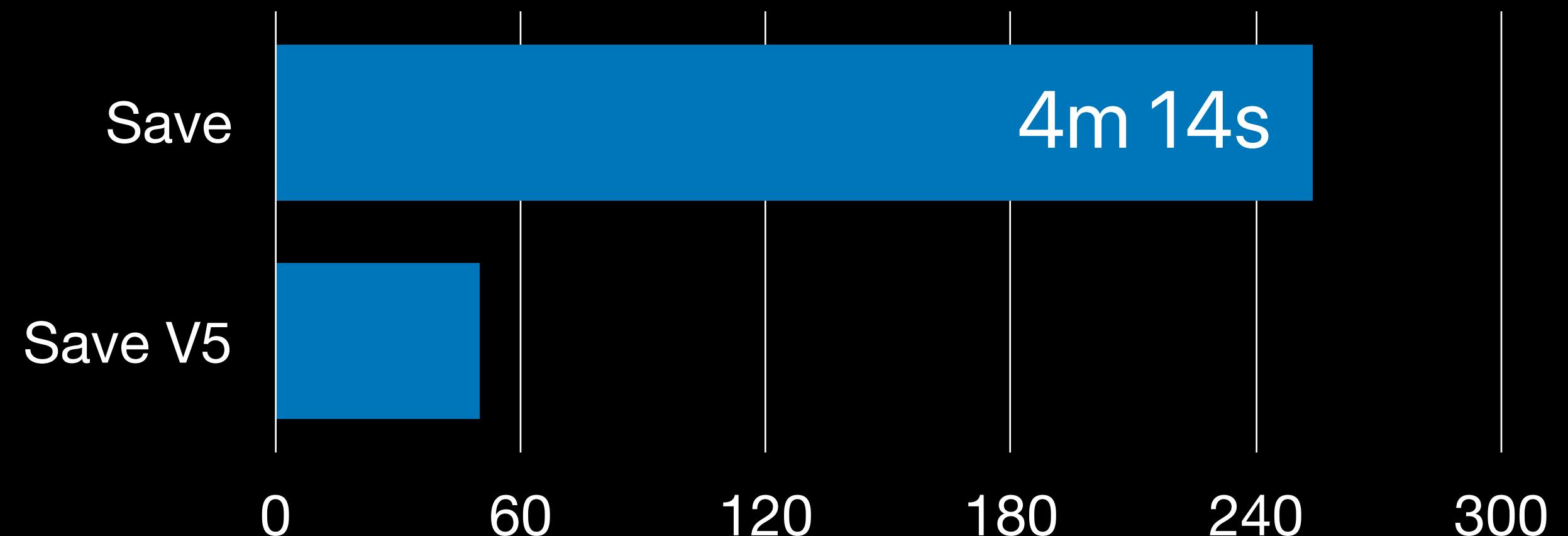


# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

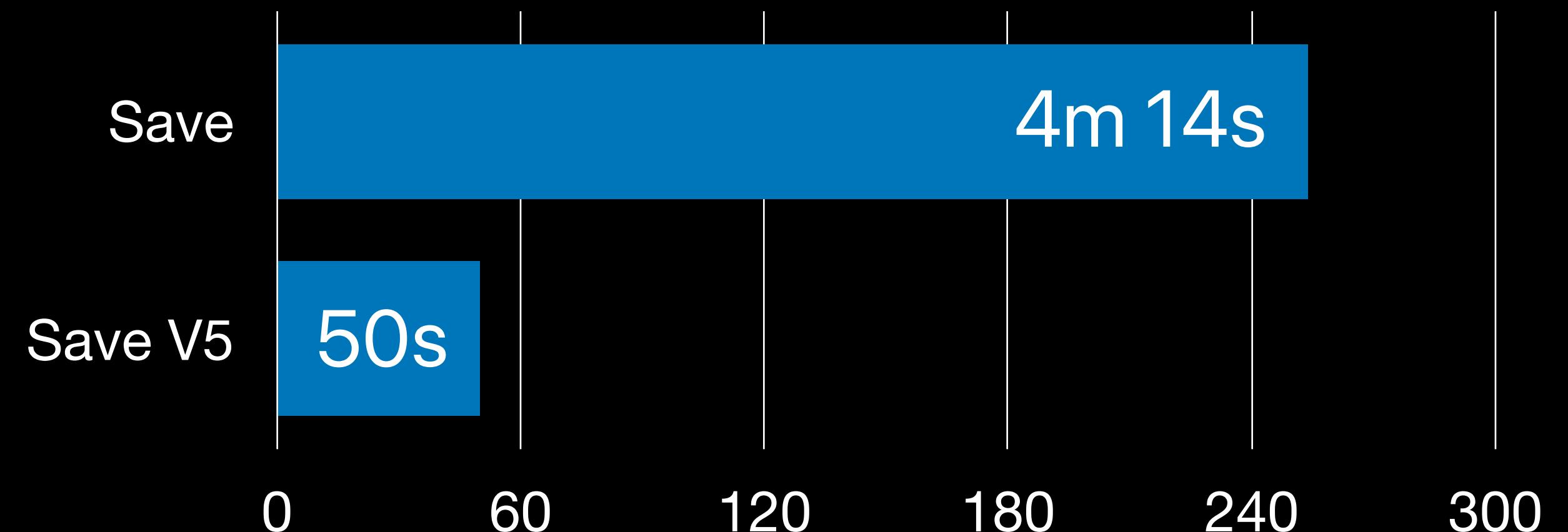


# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files



# Improving save time

# Improving save time

Pitfalls:

# Improving save time

## Pitfalls:

- sqlite fragmentation

# Improving save time

## Pitfalls:

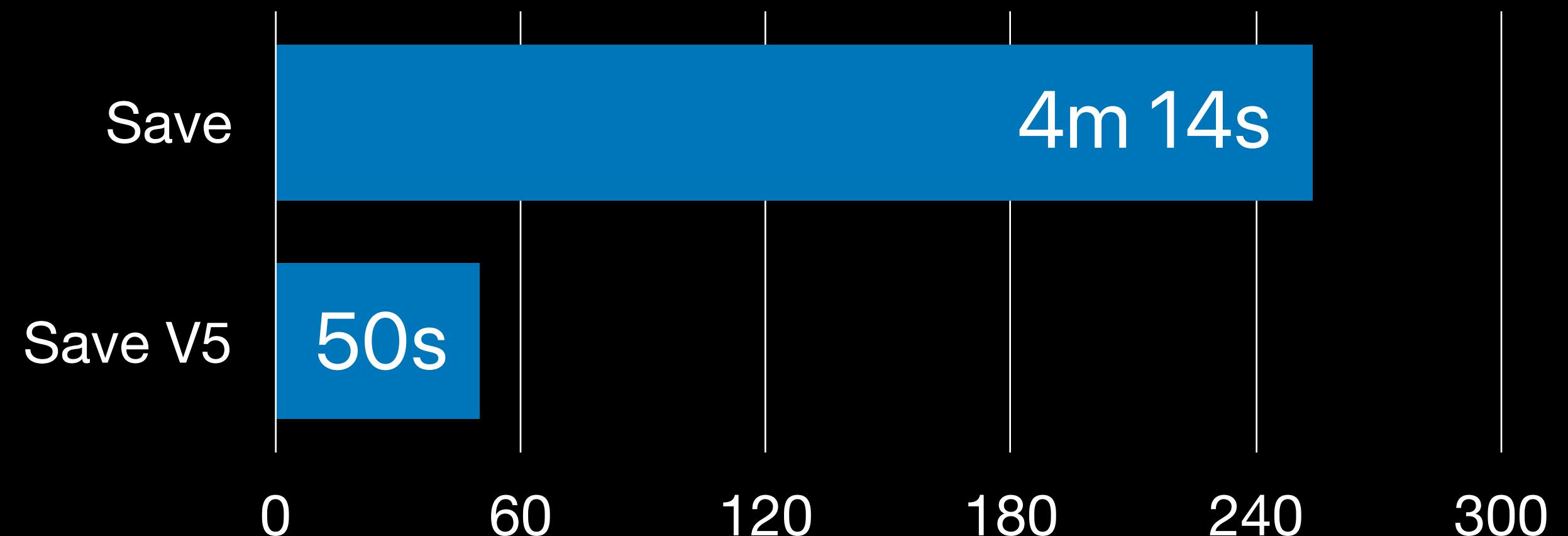
- sqlite fragmentation
- APFS copy-on-write fragmentation

# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

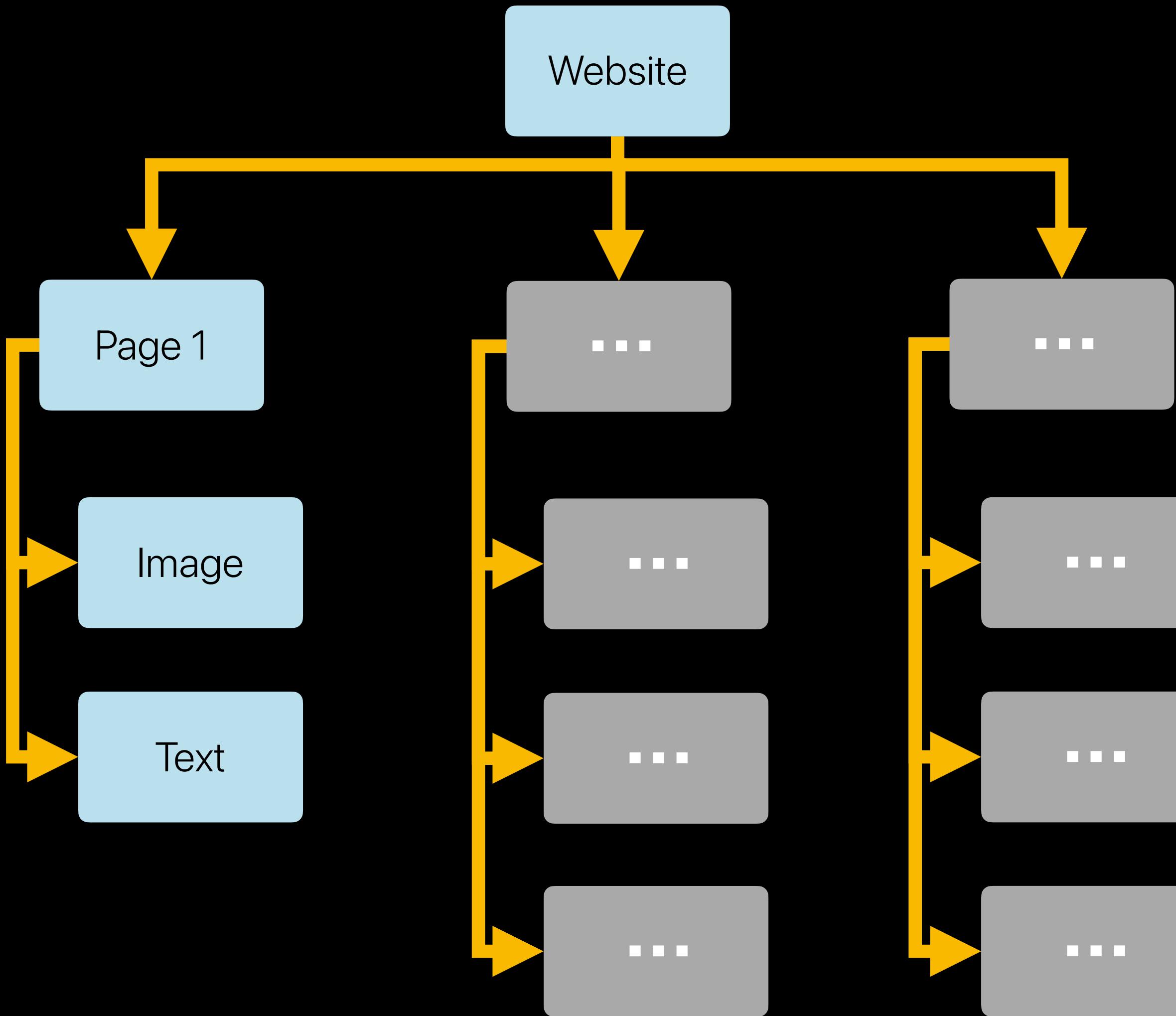


# Improving save time

- Don't do unnecessary work

# Improving save time

## Object Store



# Improving save time



website.sparkle

Website

Page 1

Image

Text

...

...

...

...

...

...



clone.sparkle

# Improving save time



website.sparkle

Website

Page 1

Image

Text

...

...

...

...

...

...



clone.sparkle

# Improving save time



website.sparkle

Website

Page 1

Image

Text

Page 2

Gallery

Movie

Text

Page3

Button



clone.sparkle

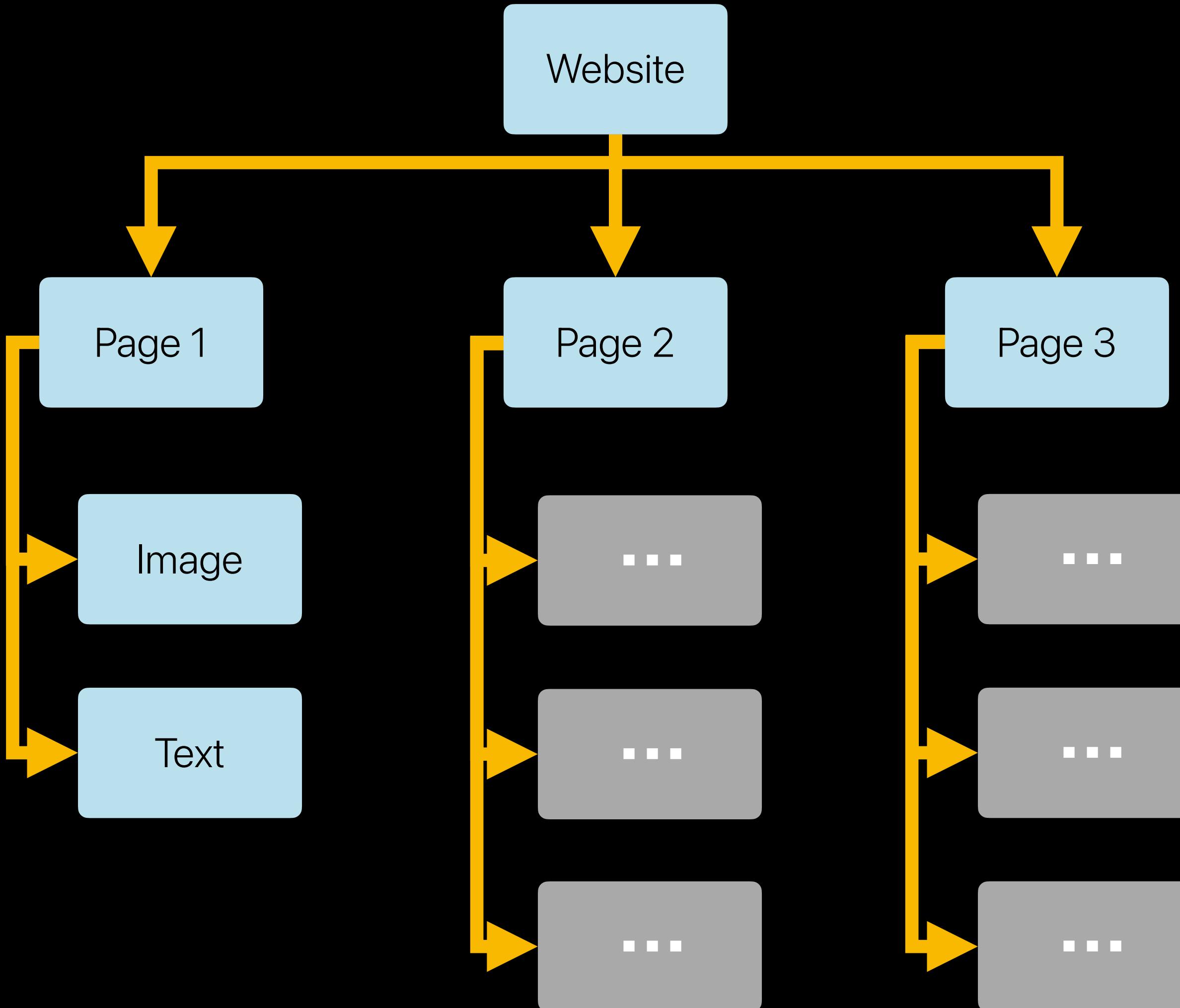
# Improving save time

- Don't do unnecessary work
- Don't load objects only to save them

Except...

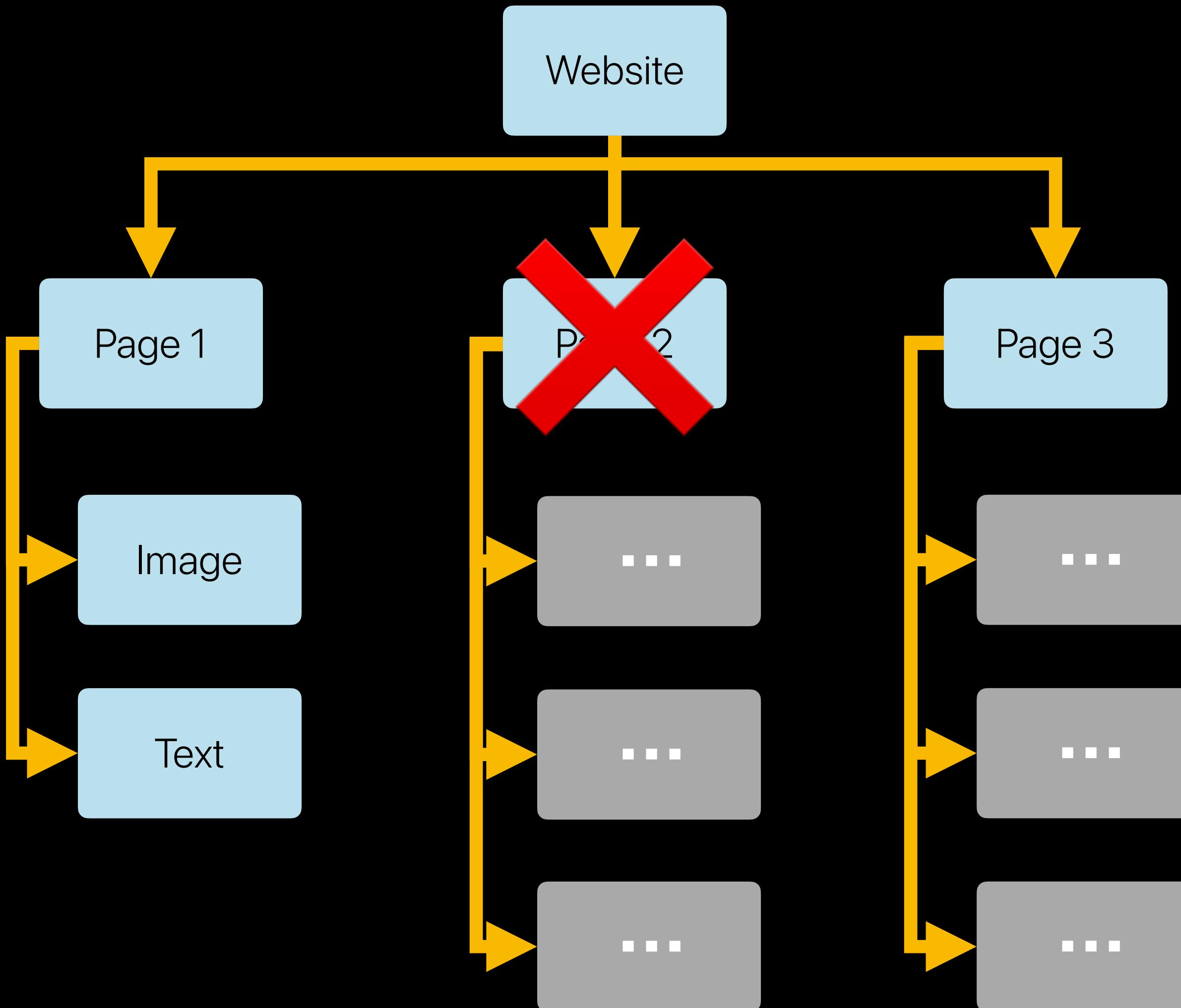
# Improving save time

## Object Store

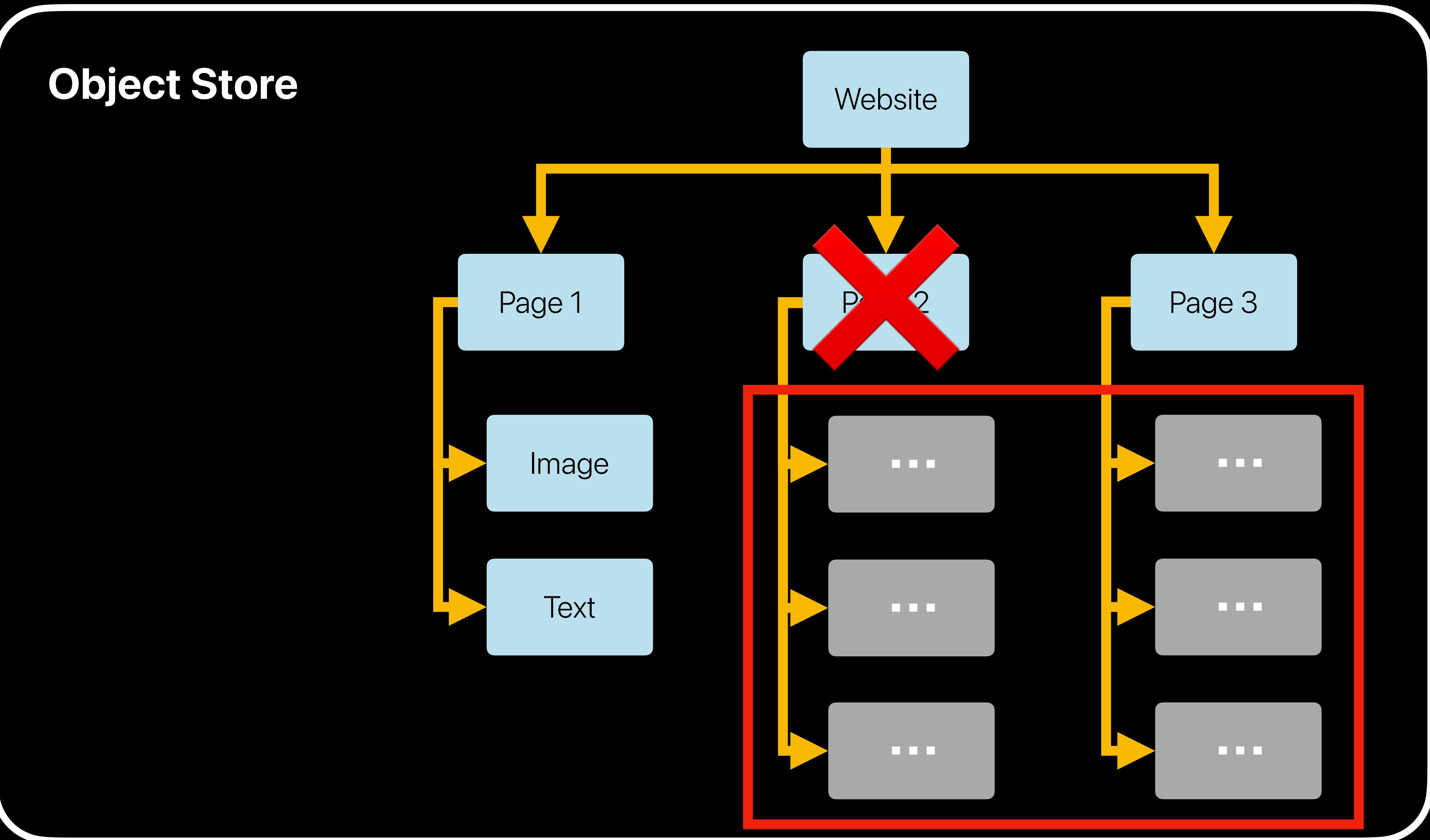


# Improving save time

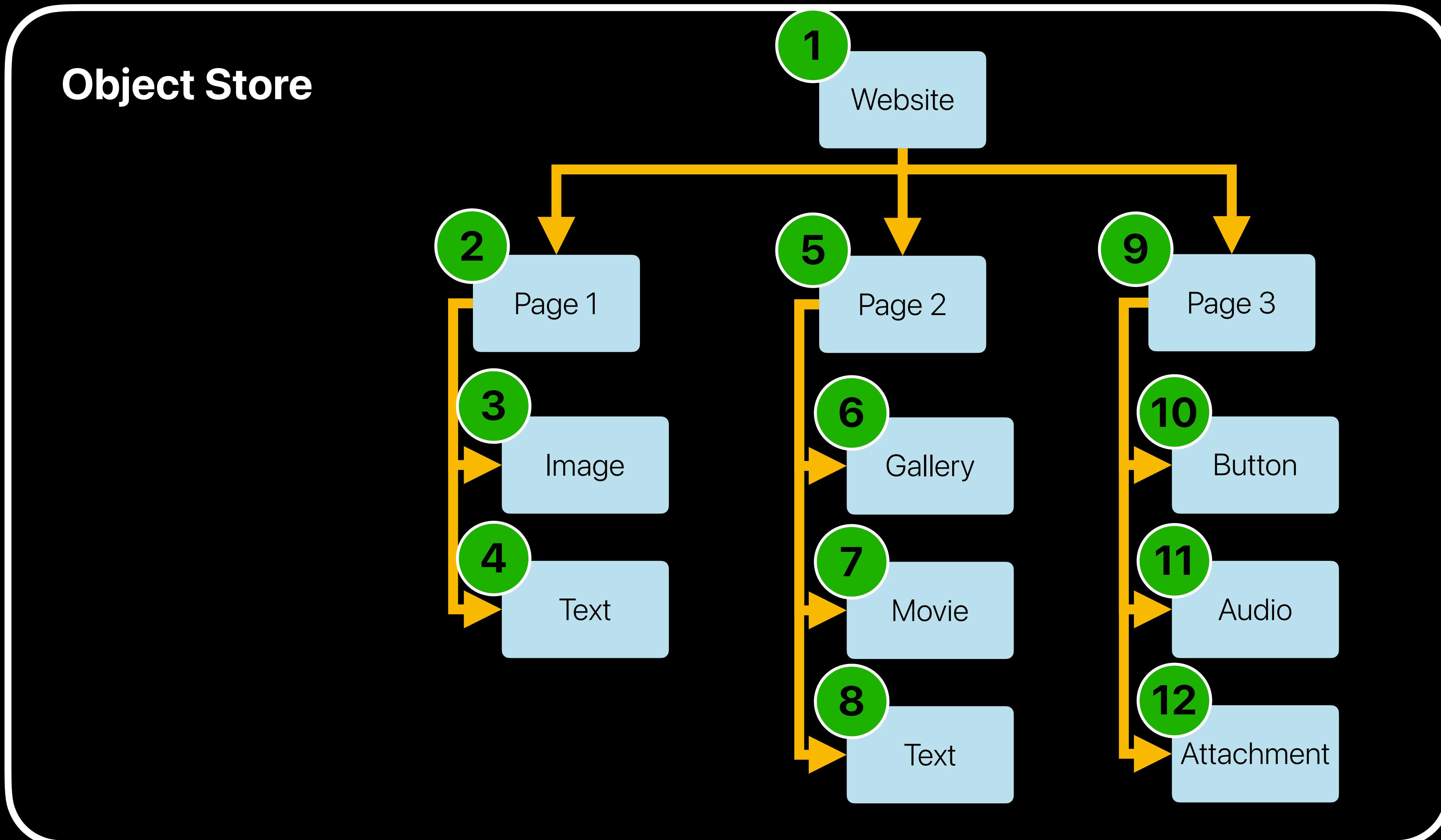
## Object Store



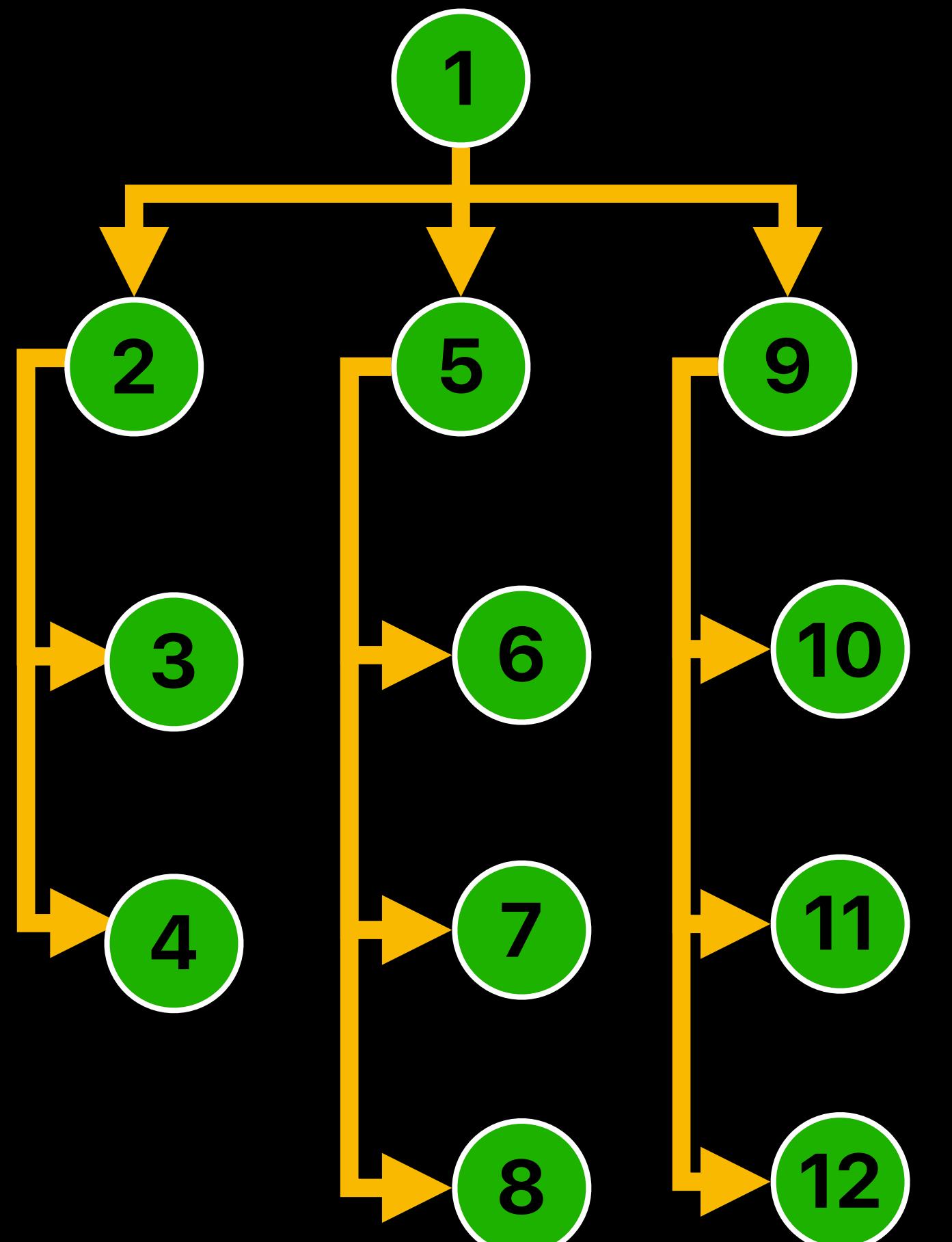
# Improving save time



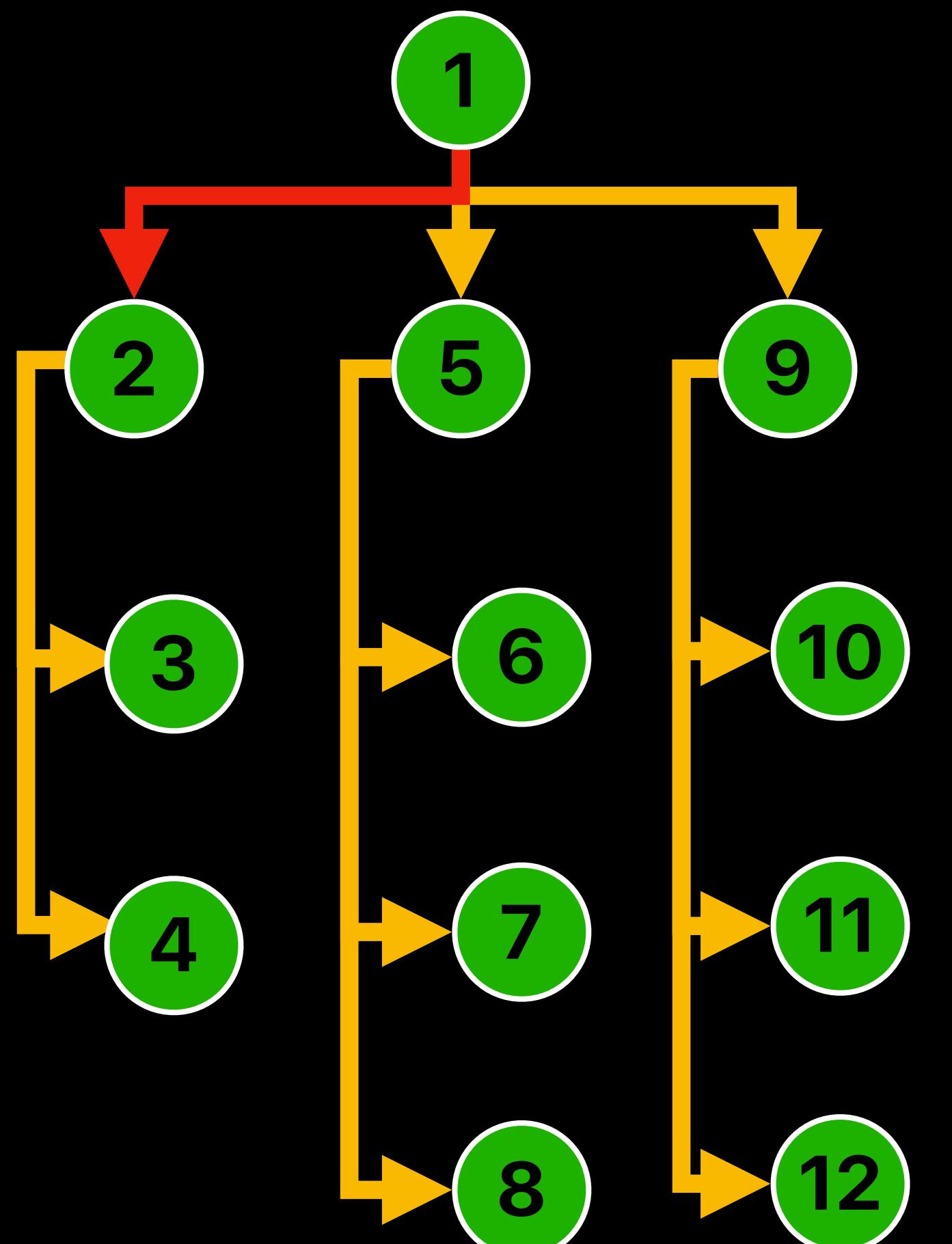
# Track live objects



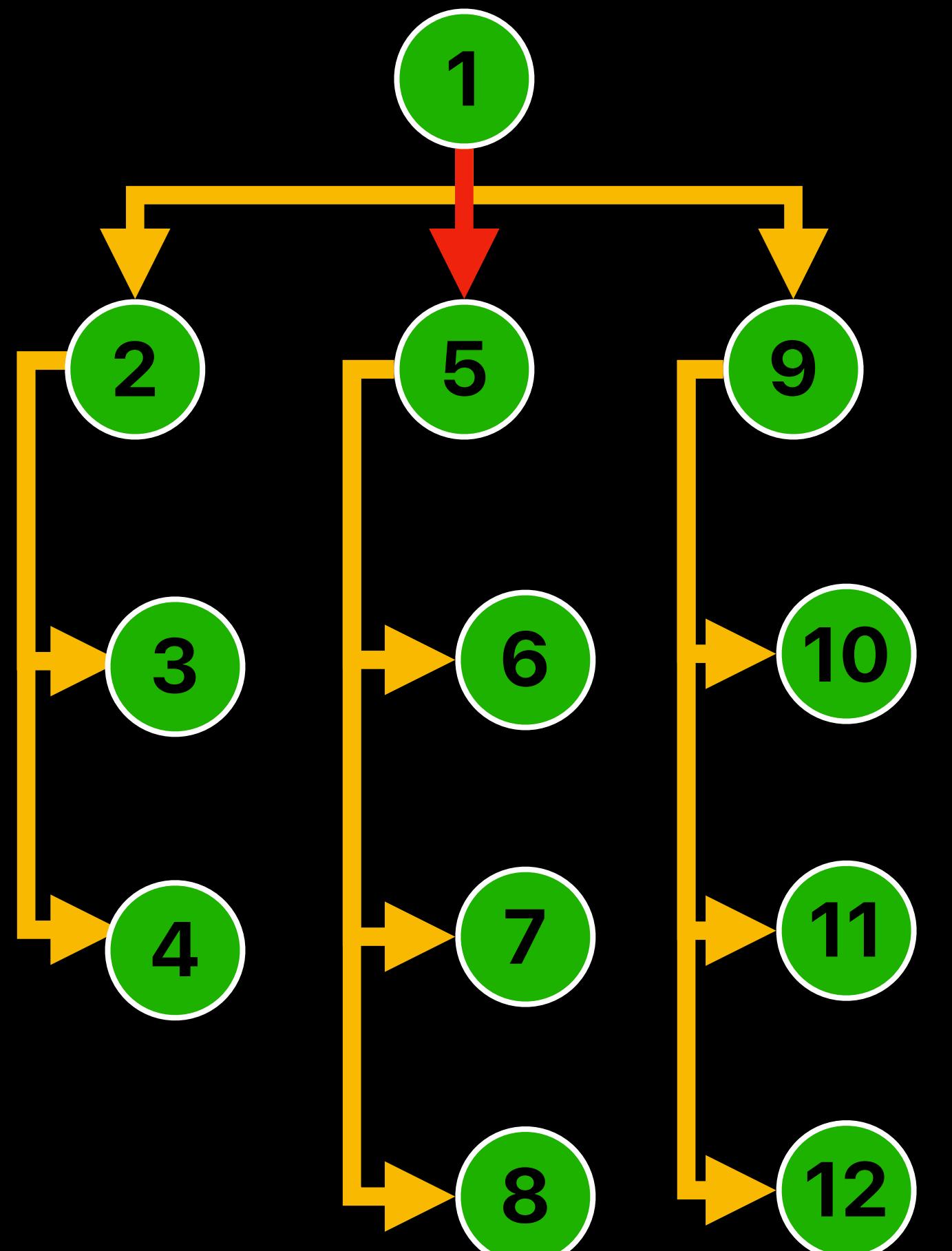
# Track live objects



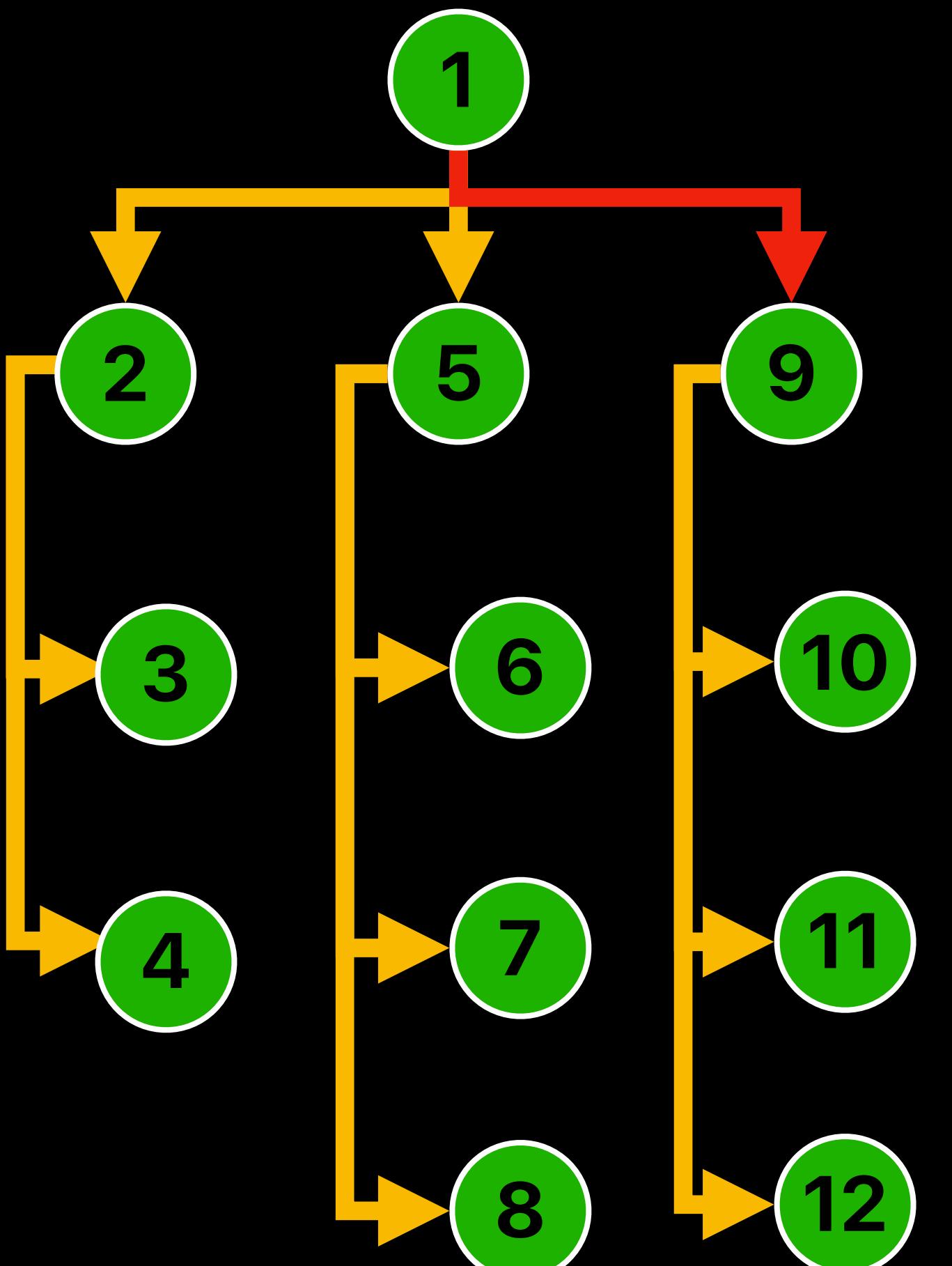
# Track live objects



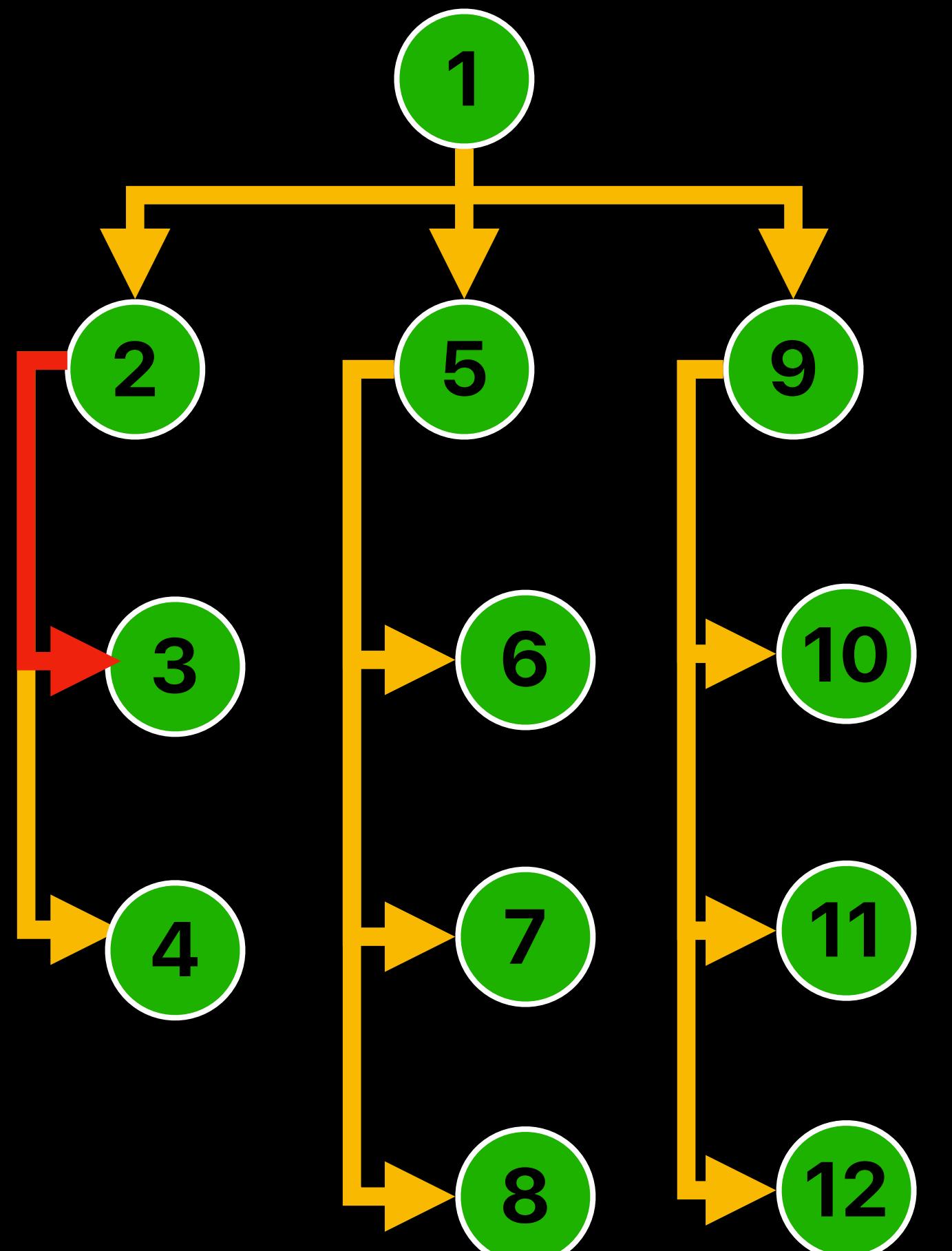
# Track live objects



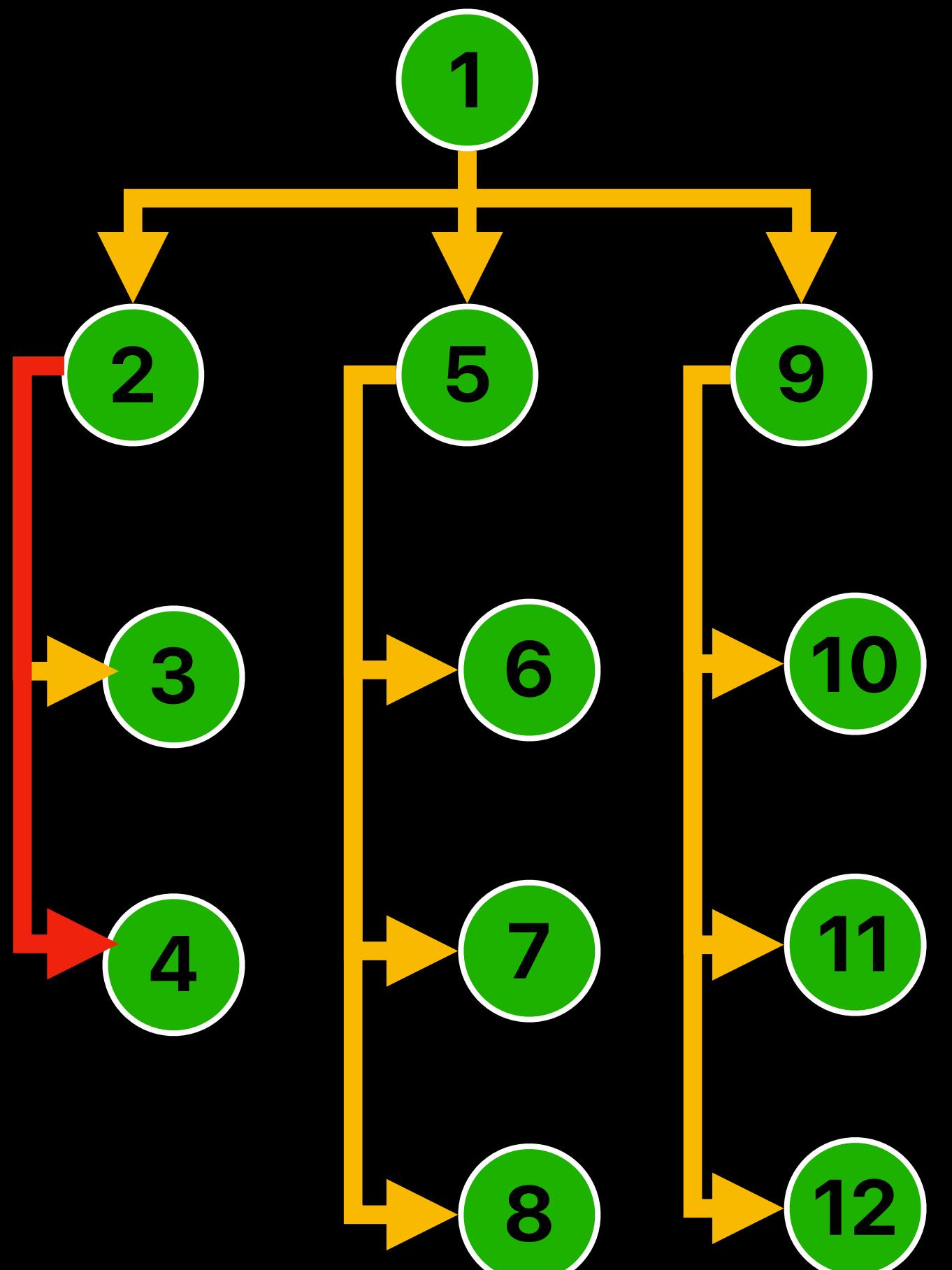
# Track live objects



# Track live objects

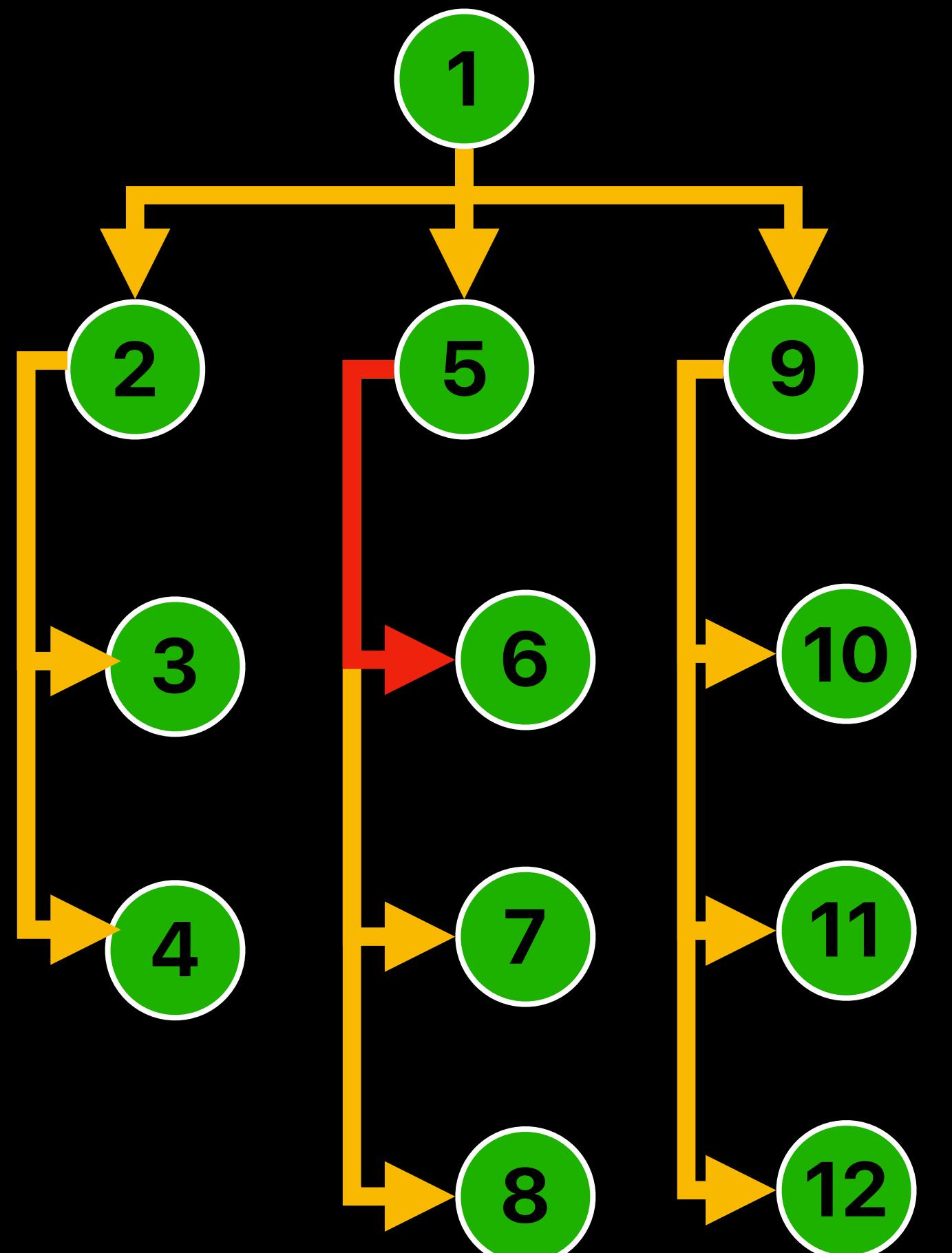


# Track live objects



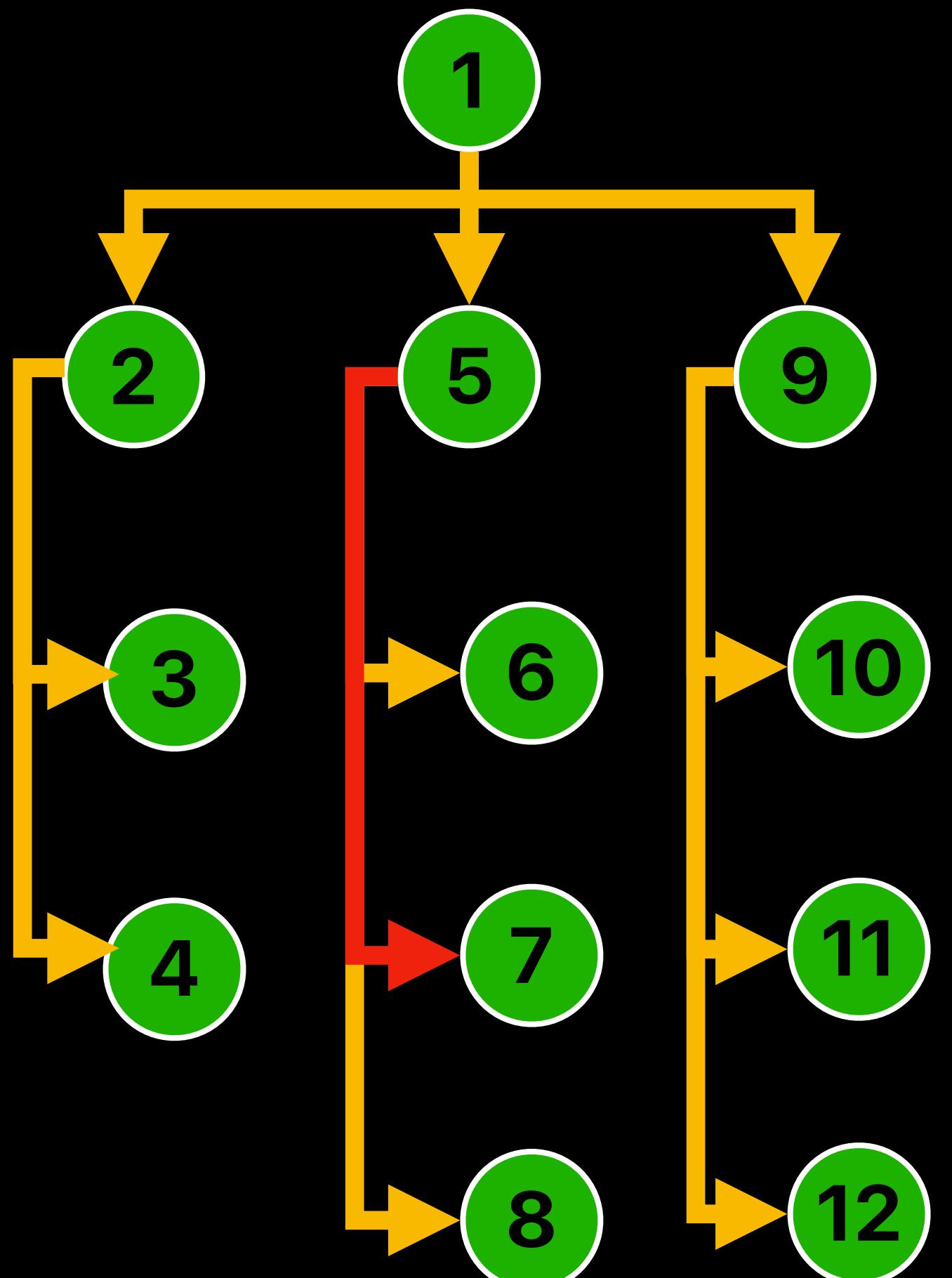
1	2
1	5
1	9
2	3
2	4

# Track live objects



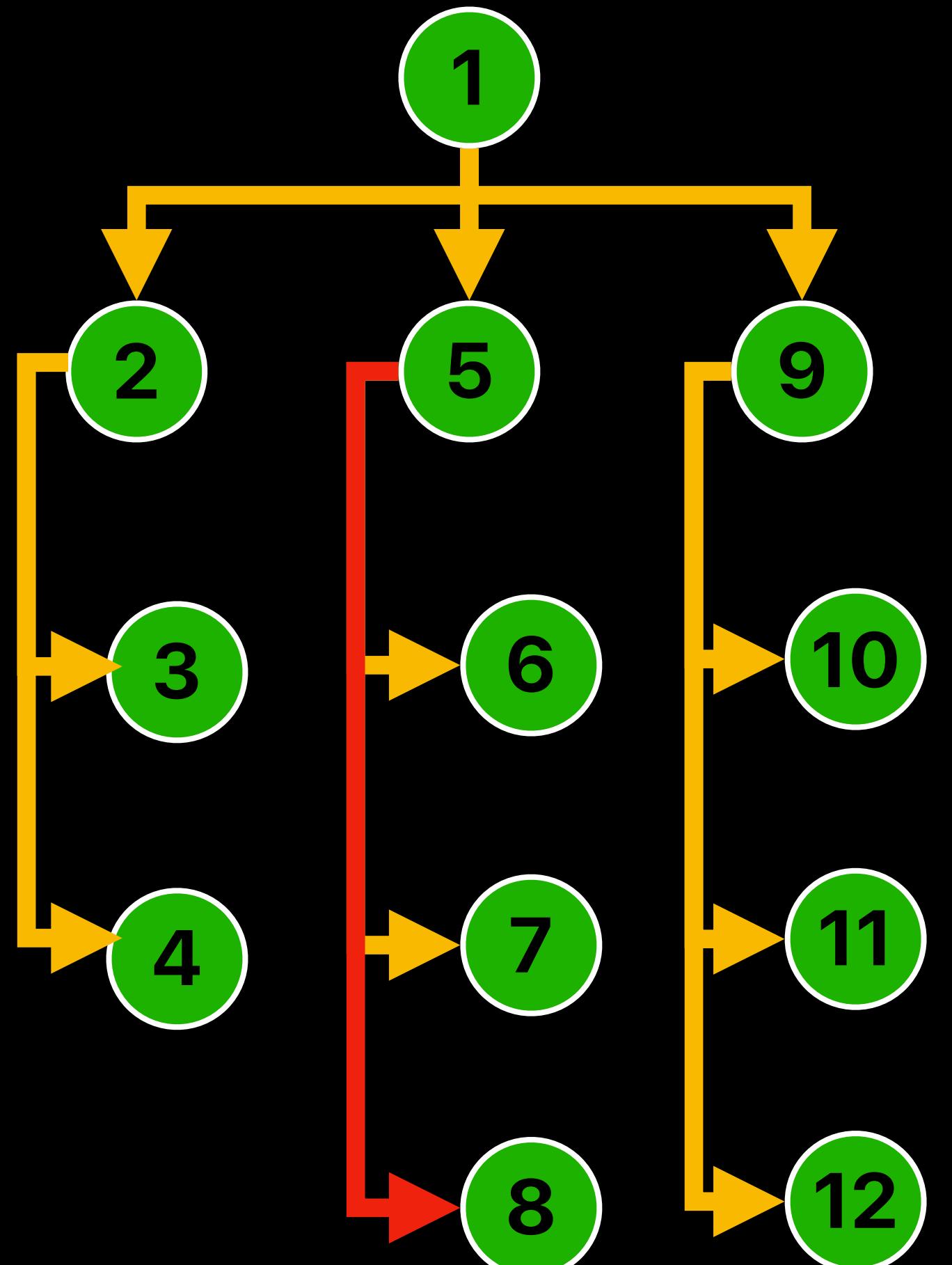
1	2
1	5
1	9
2	3
2	4
5	6

# Track live objects



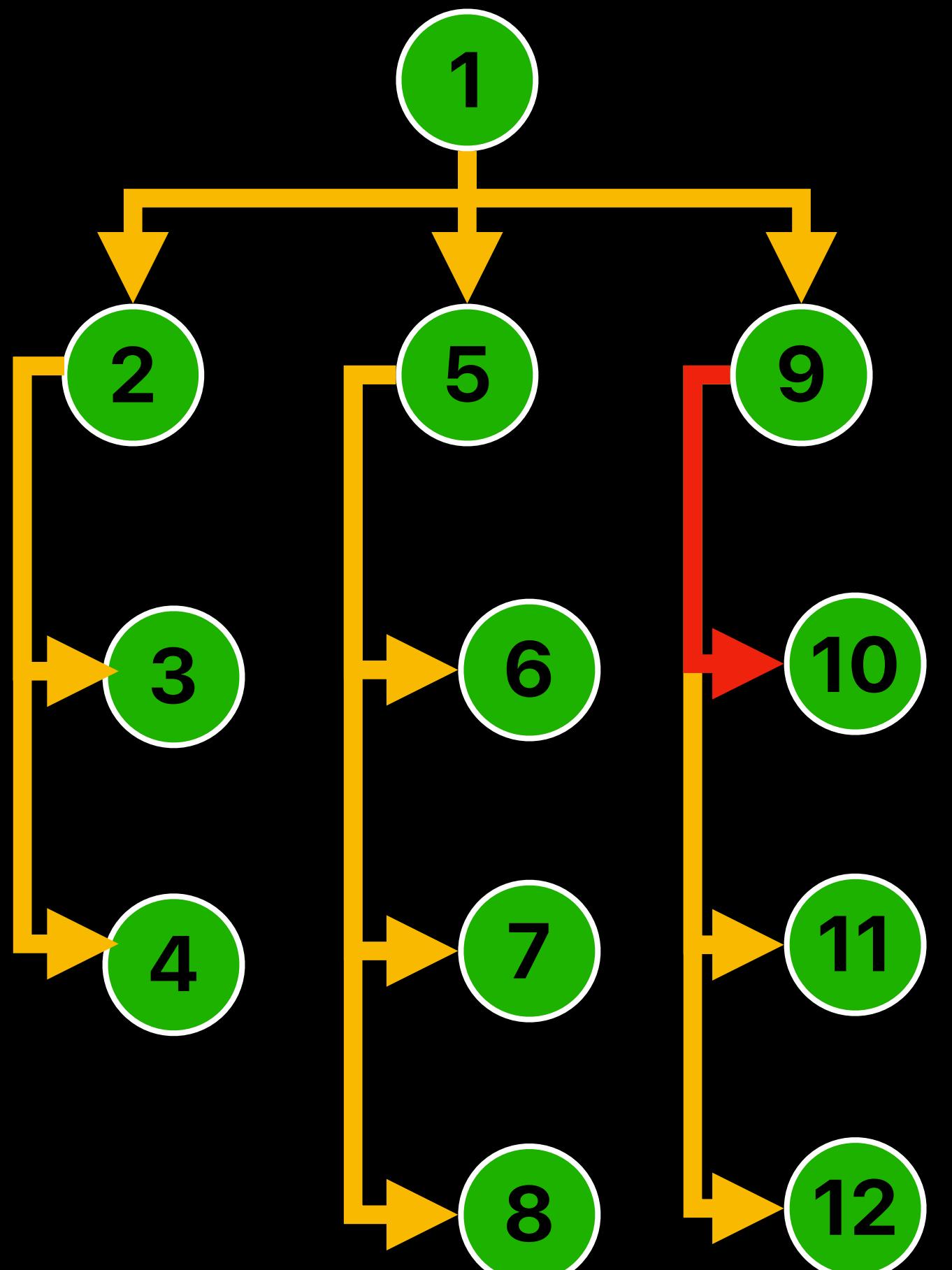
1	2
1	5
1	9
2	3
2	4
5	6
5	7
6	8
6	10
7	11
7	12
10	11
10	12
11	12

# Track live objects



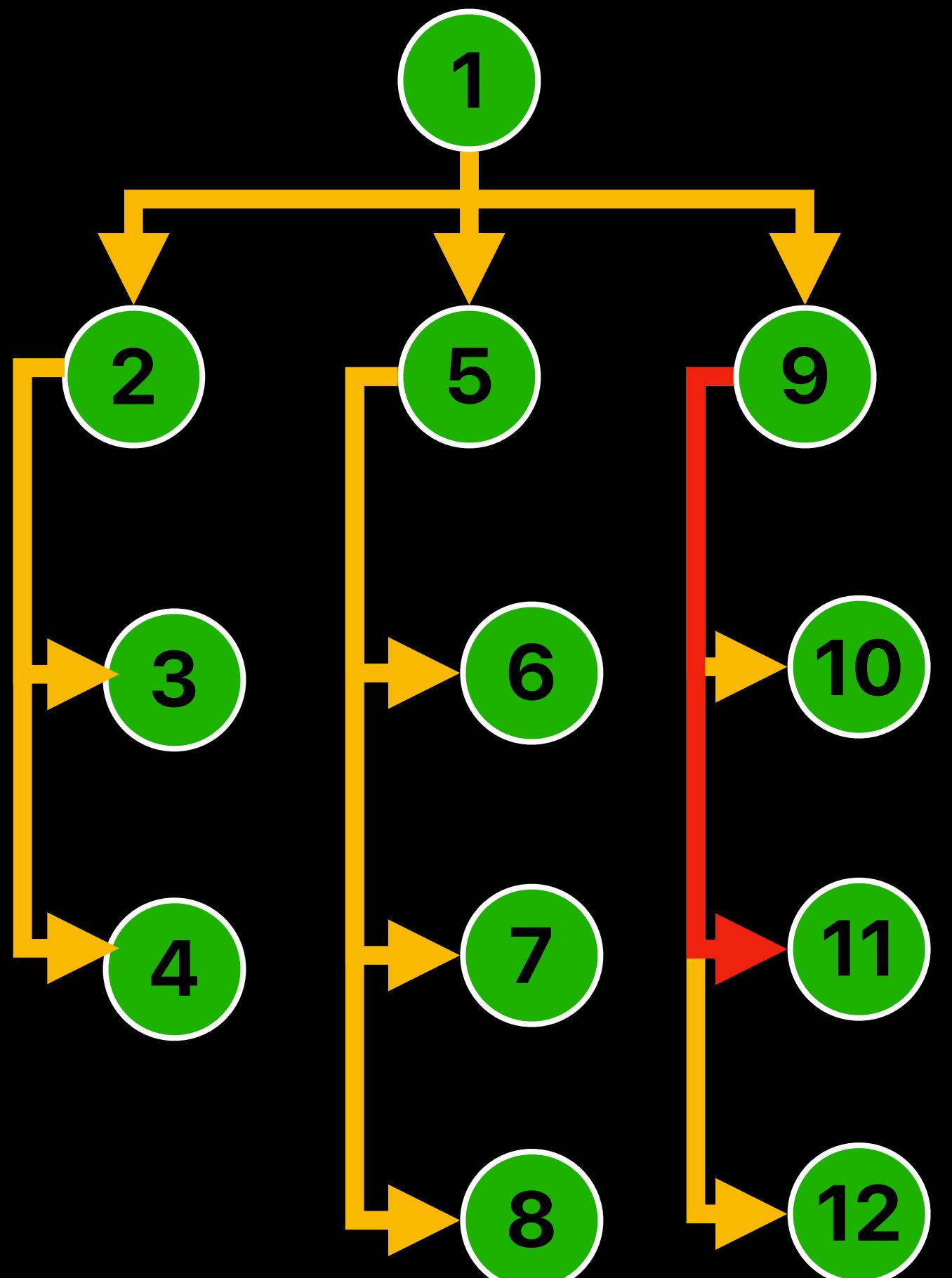
1	2
1	5
1	9
2	3
2	4
5	6
5	7
5	8

# Track live objects



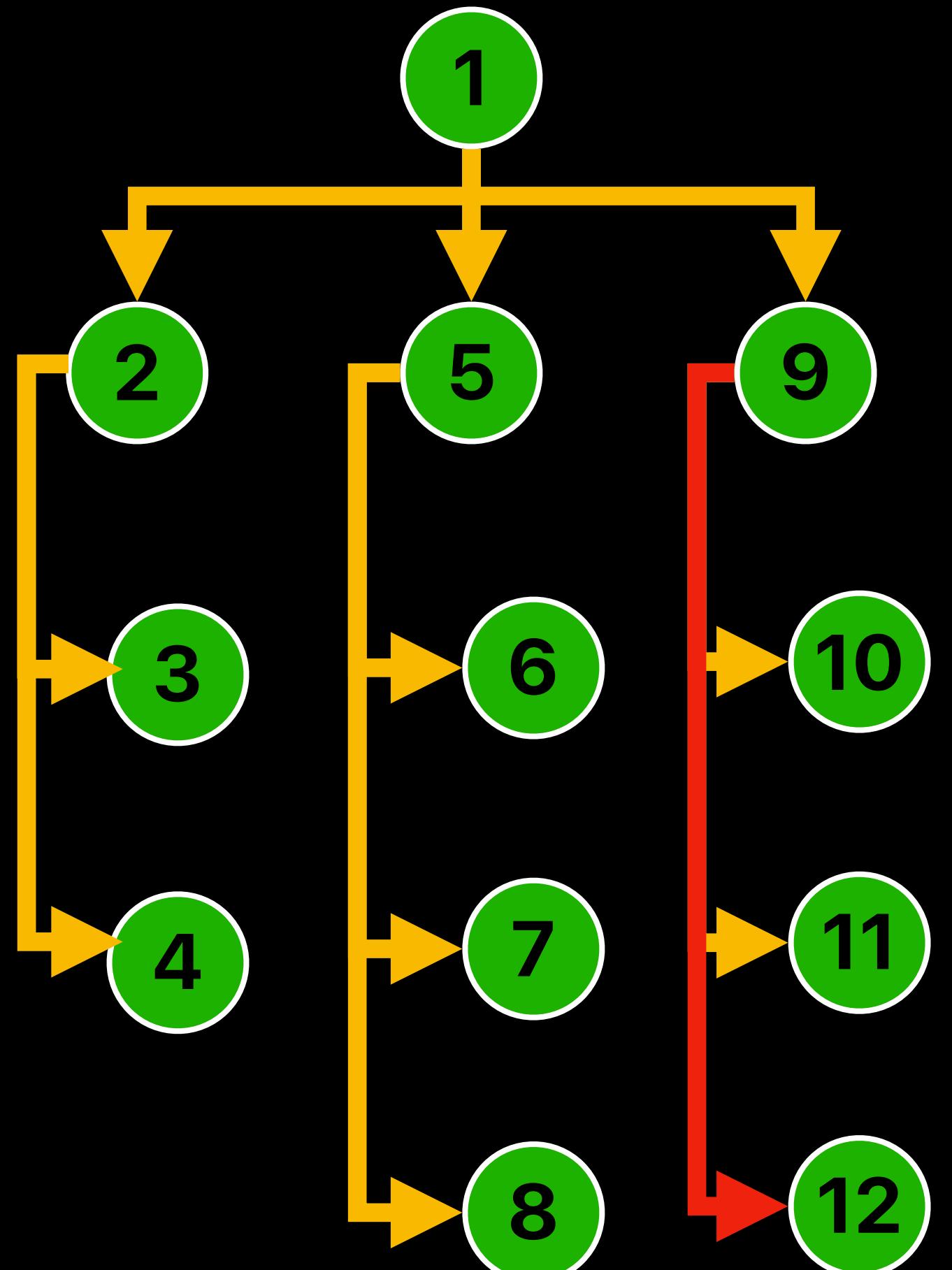
1	2
1	5
1	9
2	3
2	4
5	6
5	7
5	8
9	10

# Track live objects



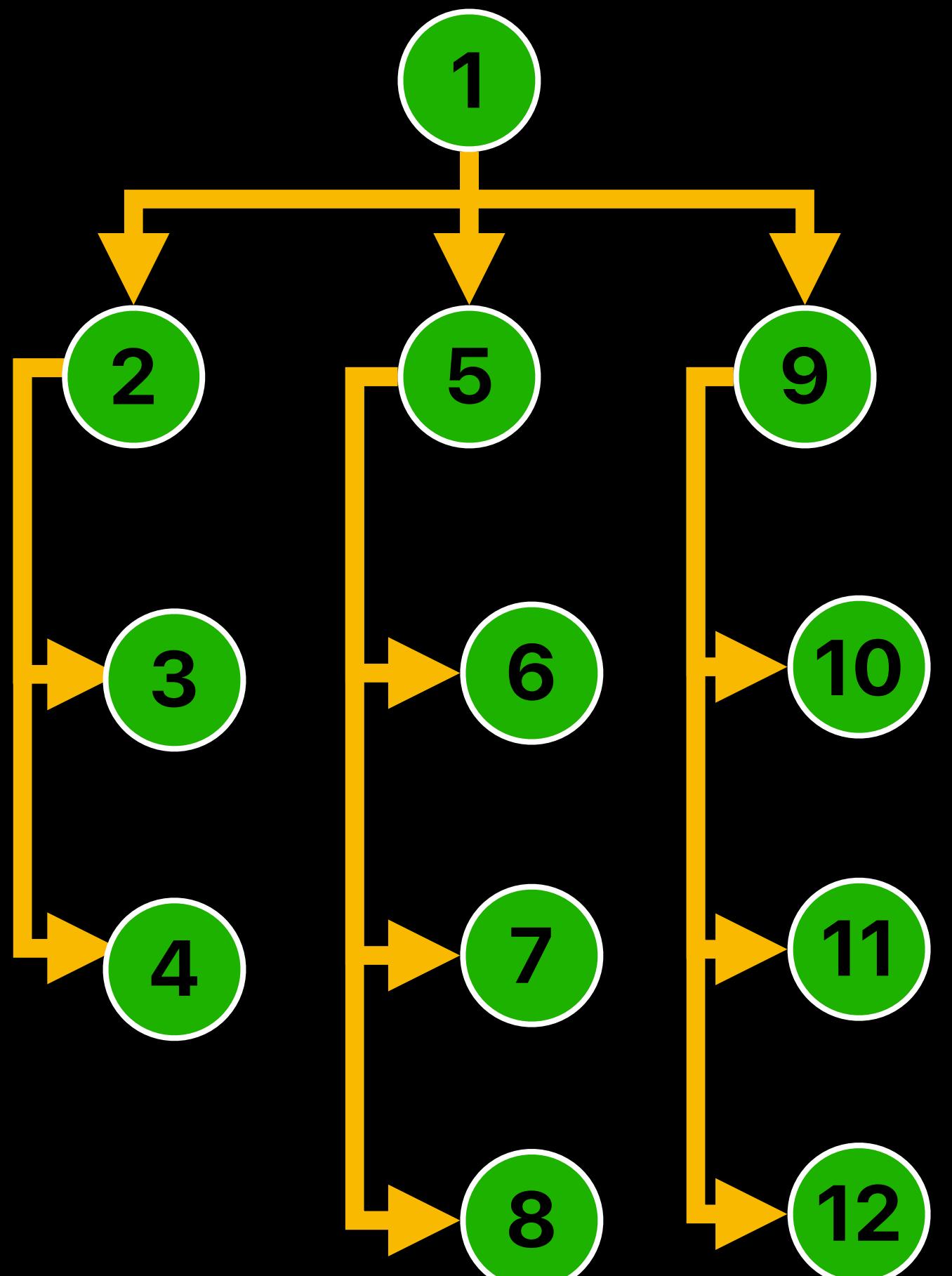
1	2
1	5
1	9
2	3
2	4
5	6
5	7
6	7
6	10
7	8
7	11
9	10
9	11

# Track live objects



1	2
1	5
1	9
2	3
2	4
5	6
5	7
5	8
9	10
9	11
9	12

# Track live objects



1	2
1	5
1	9
2	3
2	4
5	6
5	7
5	8
9	10
9	11
9	12

# Track live objects

1	2
1	5
1	9
2	3
2	4
5	6
5	7
5	8
9	10
9	11
9	12

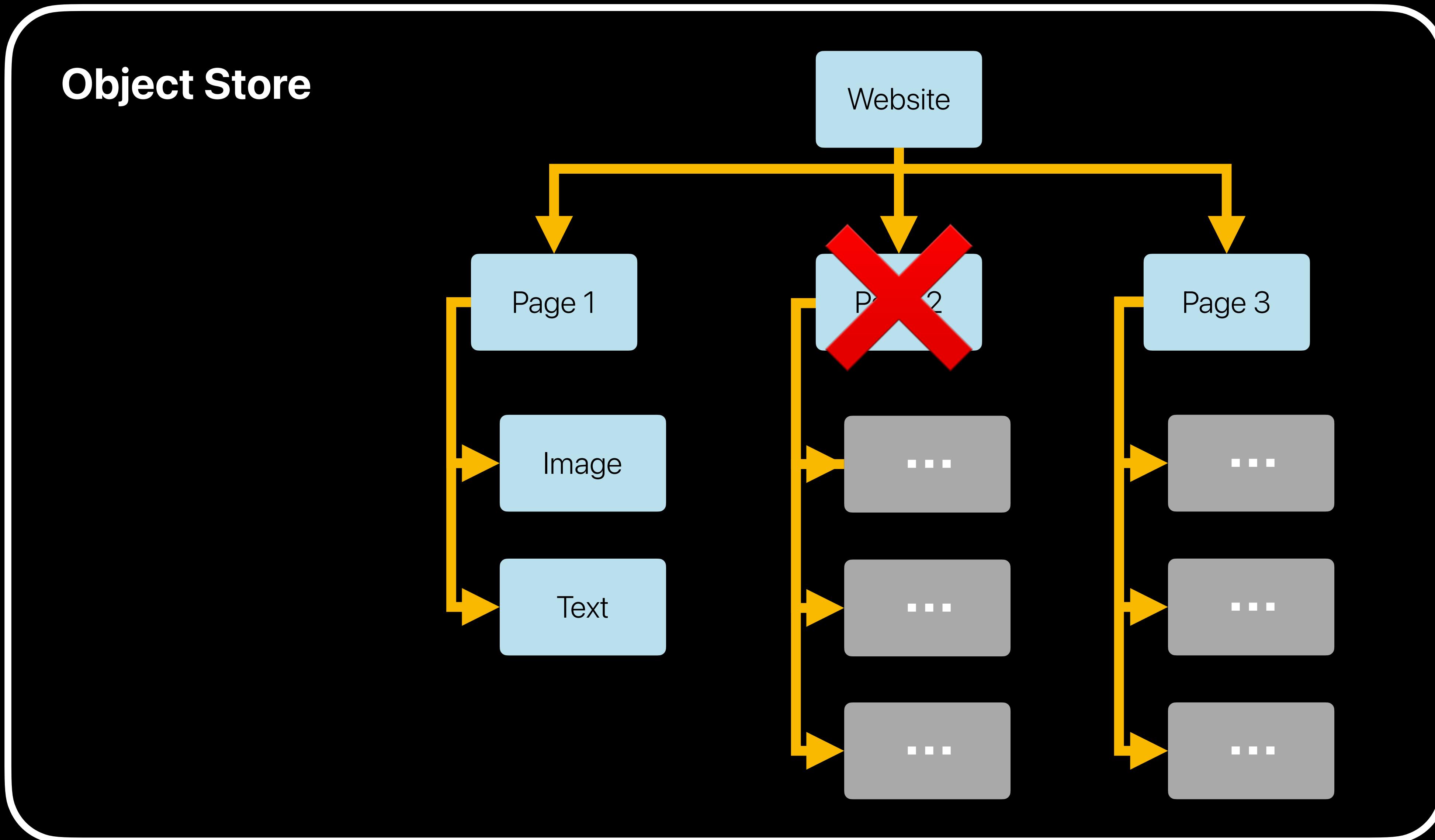
# Track live objects

i1	i2	gen
1	2	
1	5	
1	9	
2	3	
2	4	
5	6	
5	7	
5	8	
9	10	
9	11	
9	12	

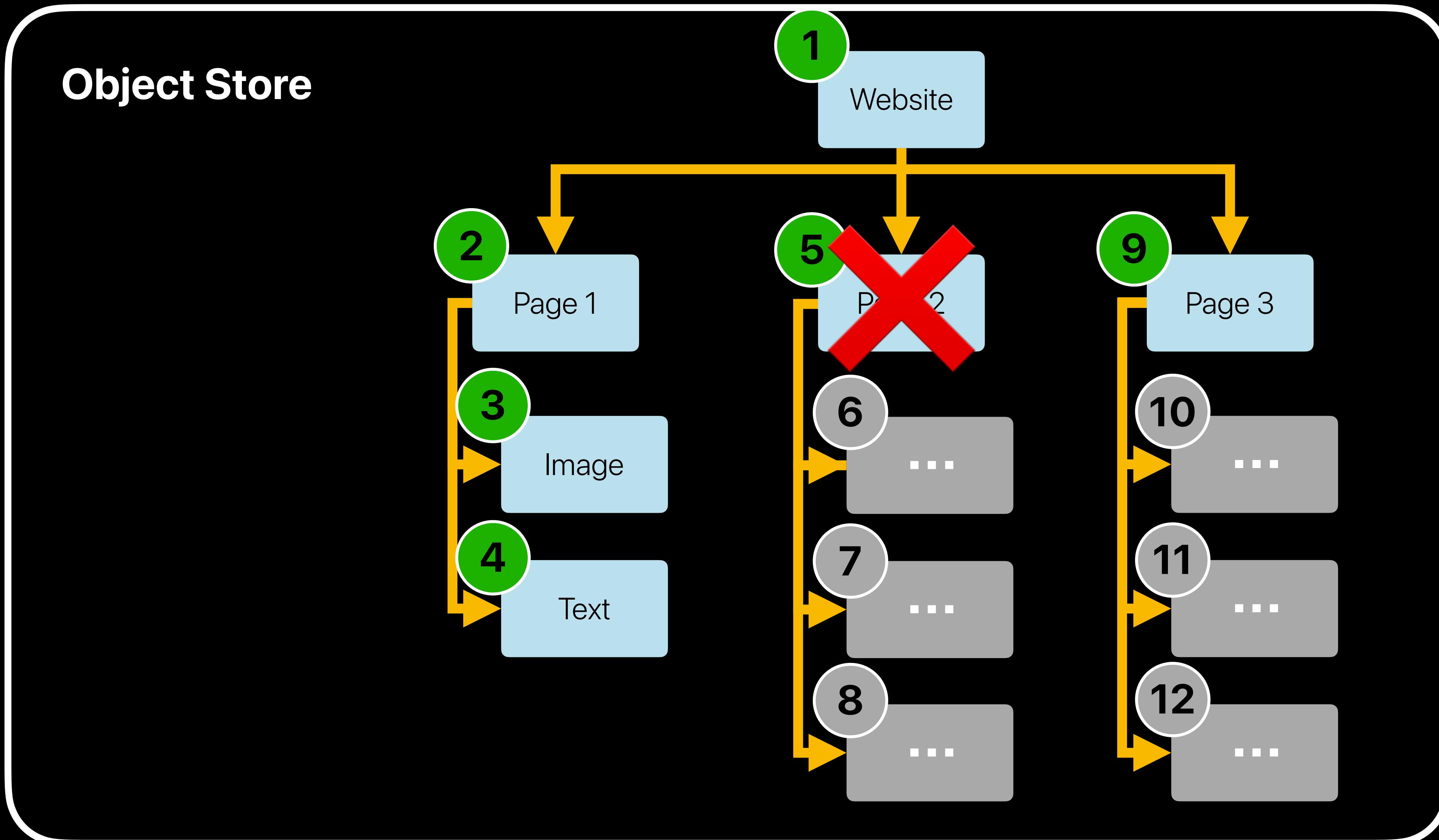
# Track live objects

i1	i2	gen
1	2	41
1	5	41
1	9	41
2	3	41
2	4	41
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

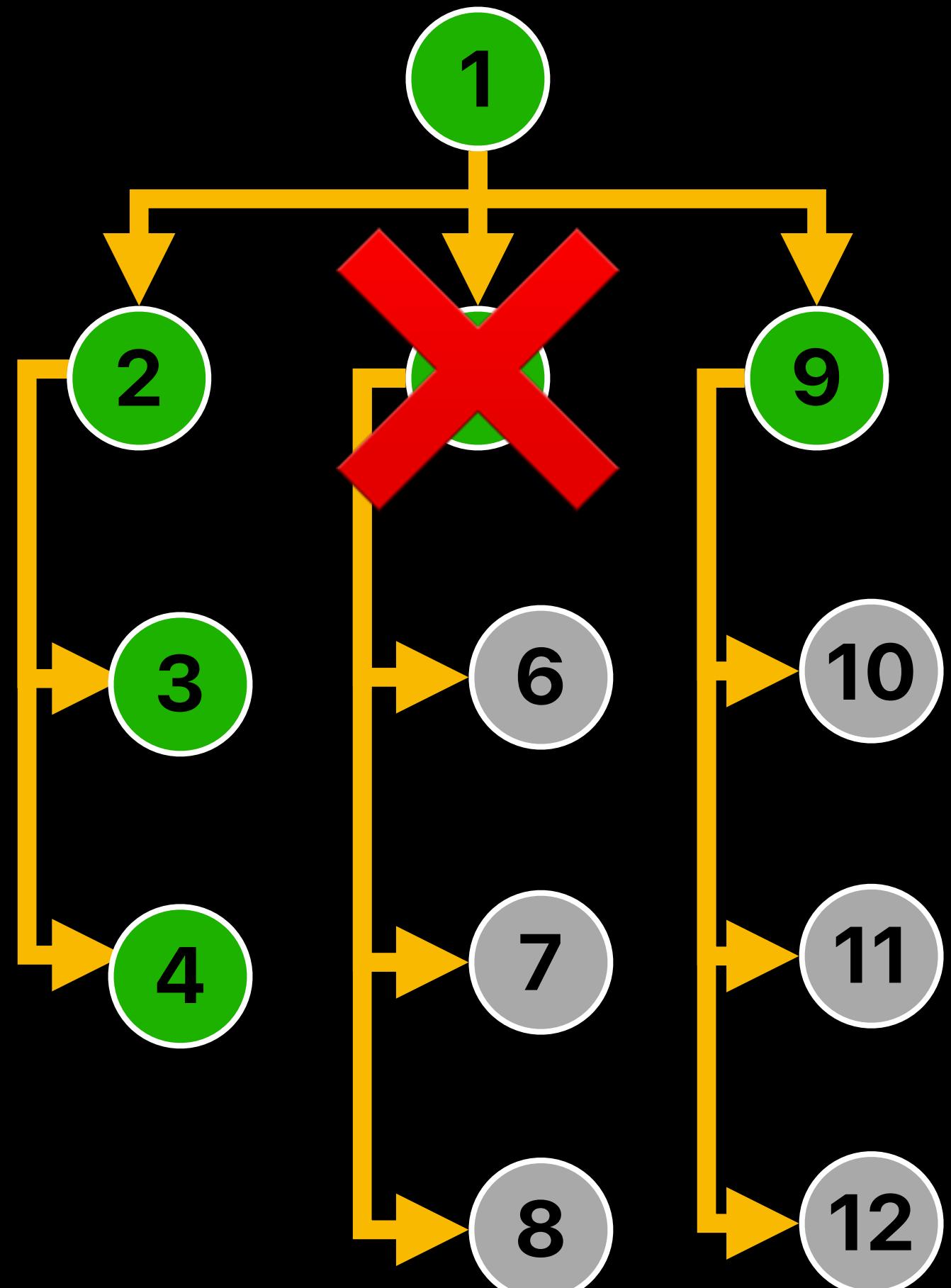
# Track live objects



# Track live objects

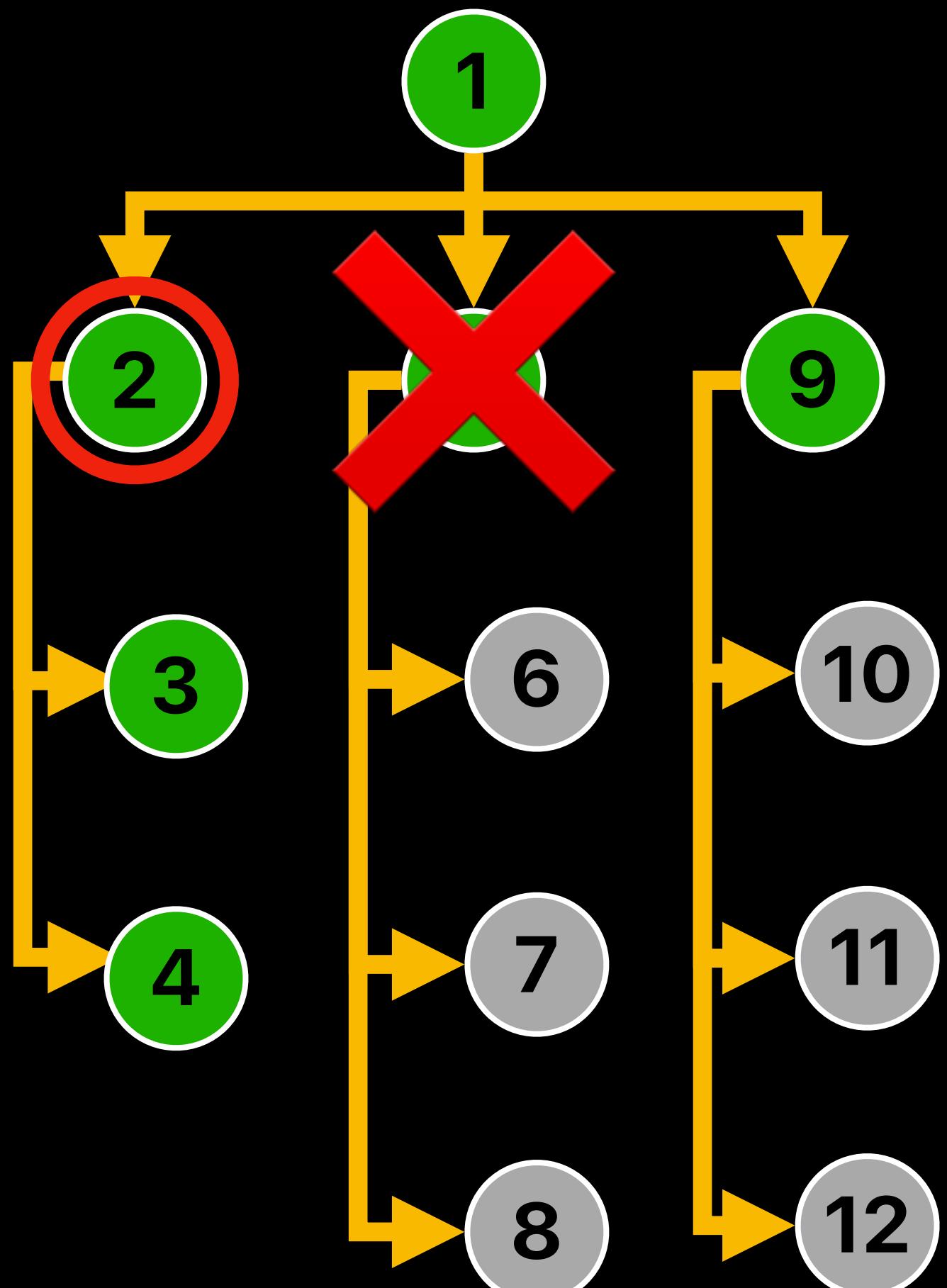


# Track live objects



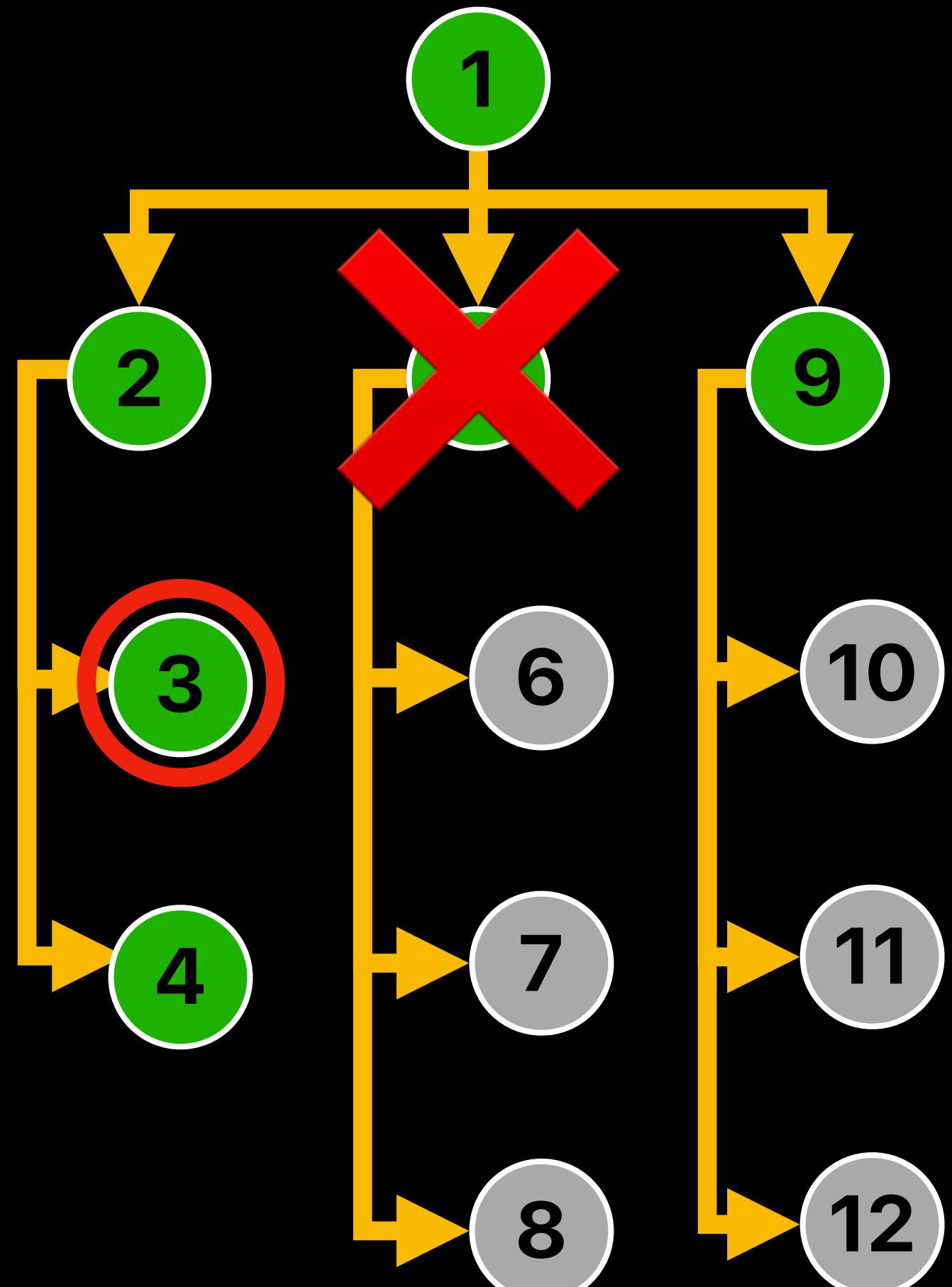
i1	i2	gen
1	2	41
1	5	41
1	9	41
2	3	41
2	4	41
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



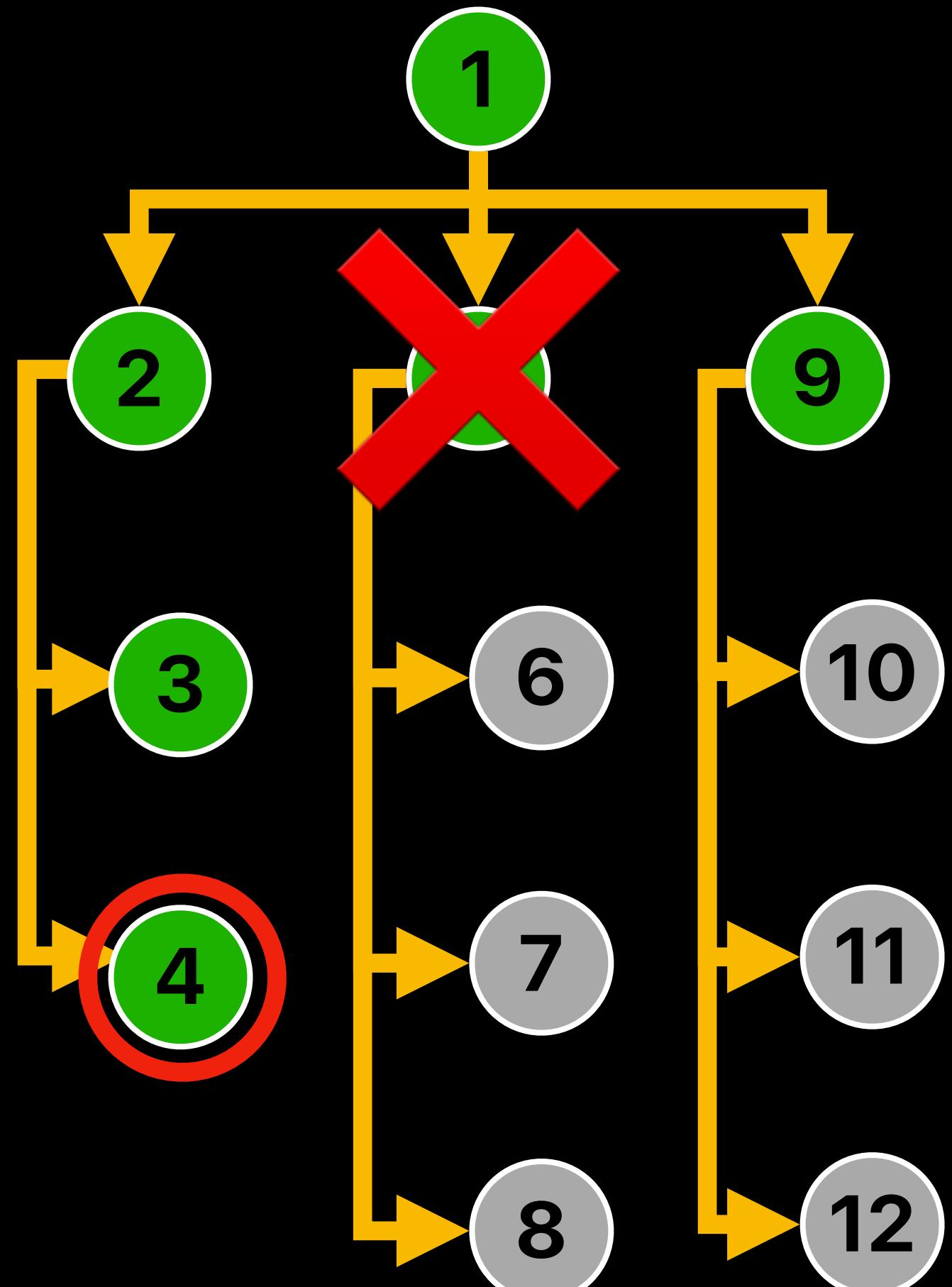
i1	i2	gen
1	2	42
1	5	41
1	9	41
2	3	41
2	4	41
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



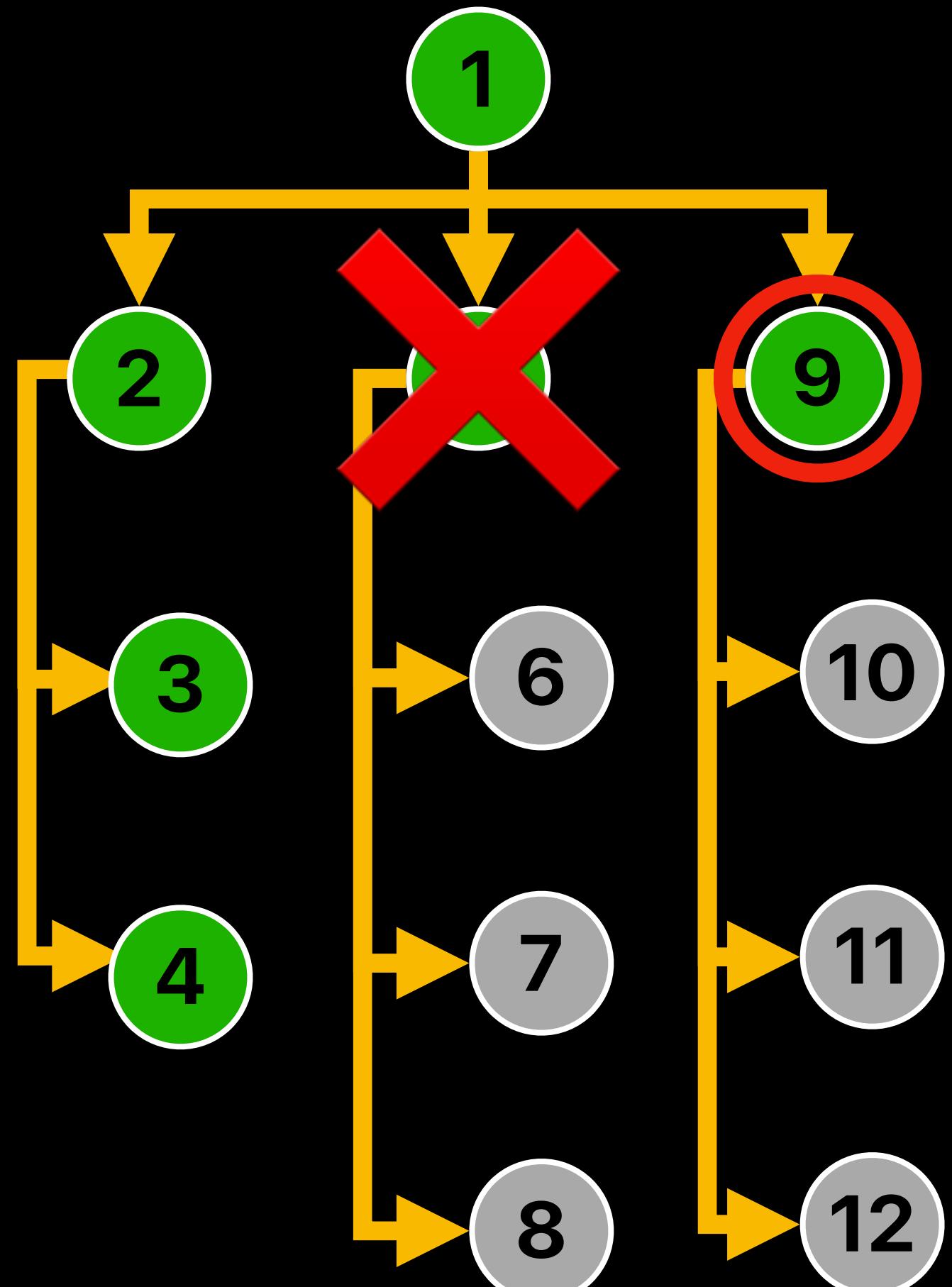
i1	i2	gen
1	2	42
1	5	41
1	9	41
2	3	42
2	4	41
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



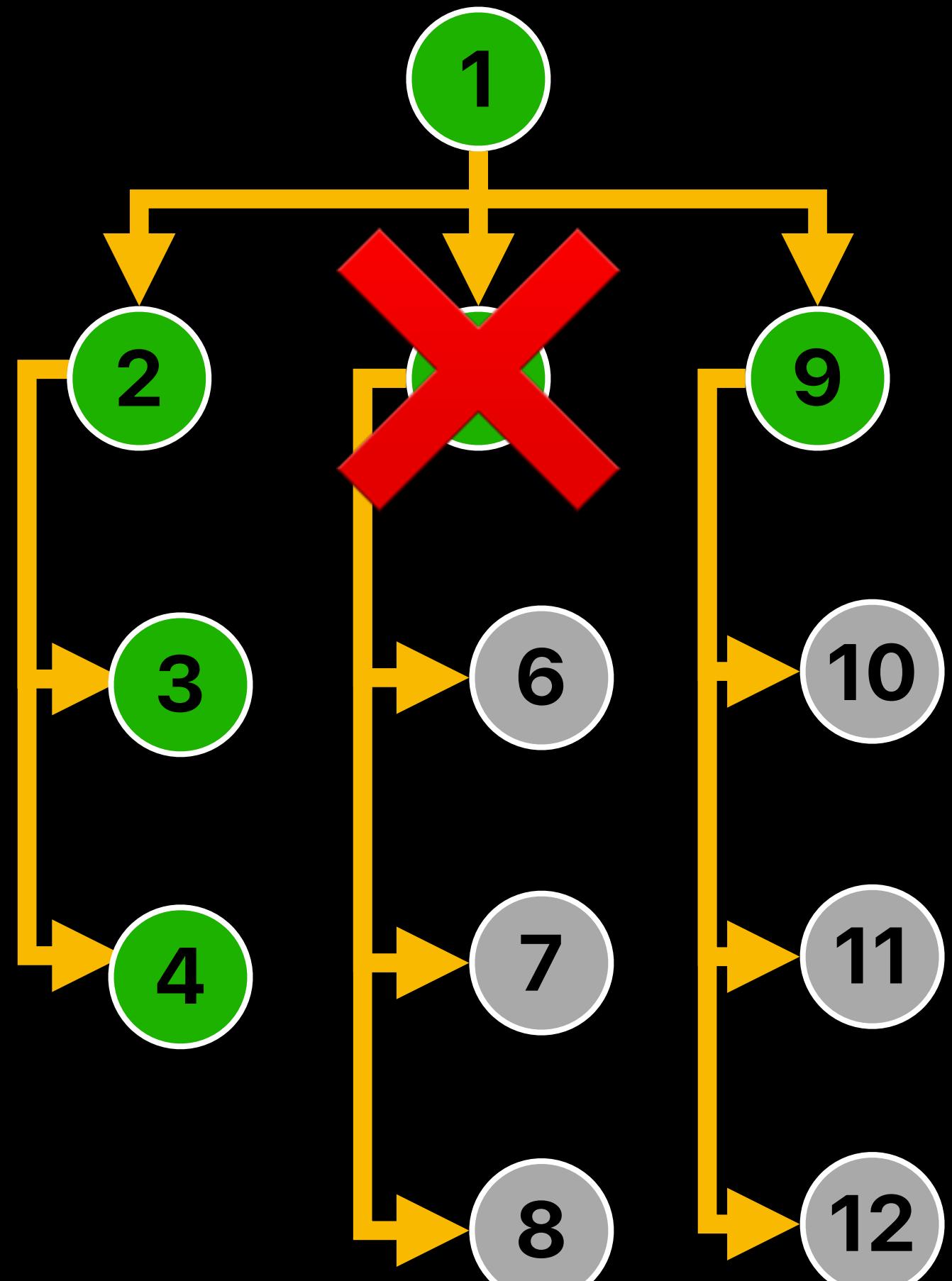
i1	i2	gen
1	2	42
1	5	41
1	9	41
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

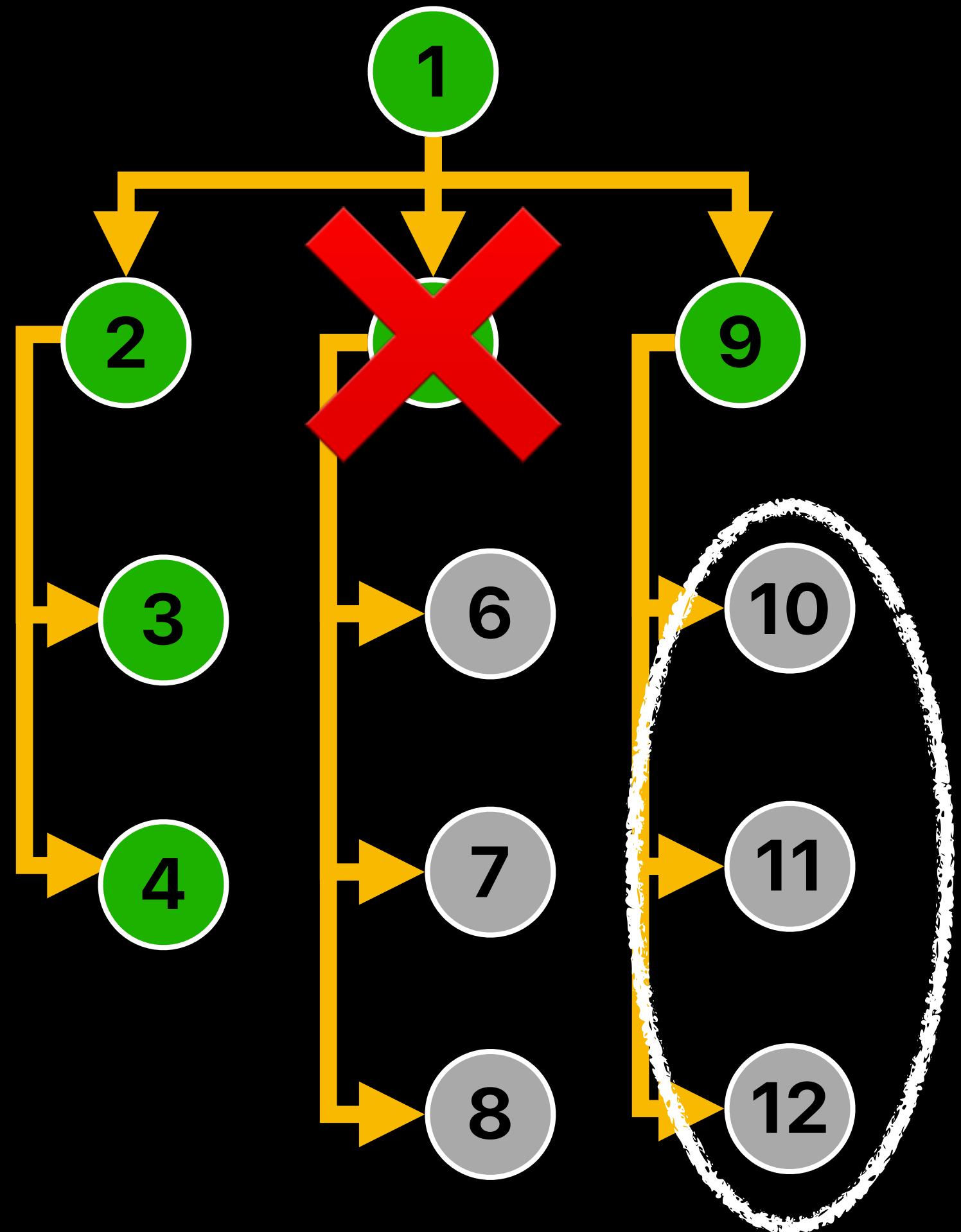
# Track live objects



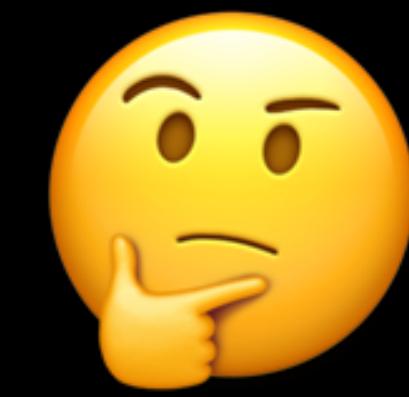
i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41



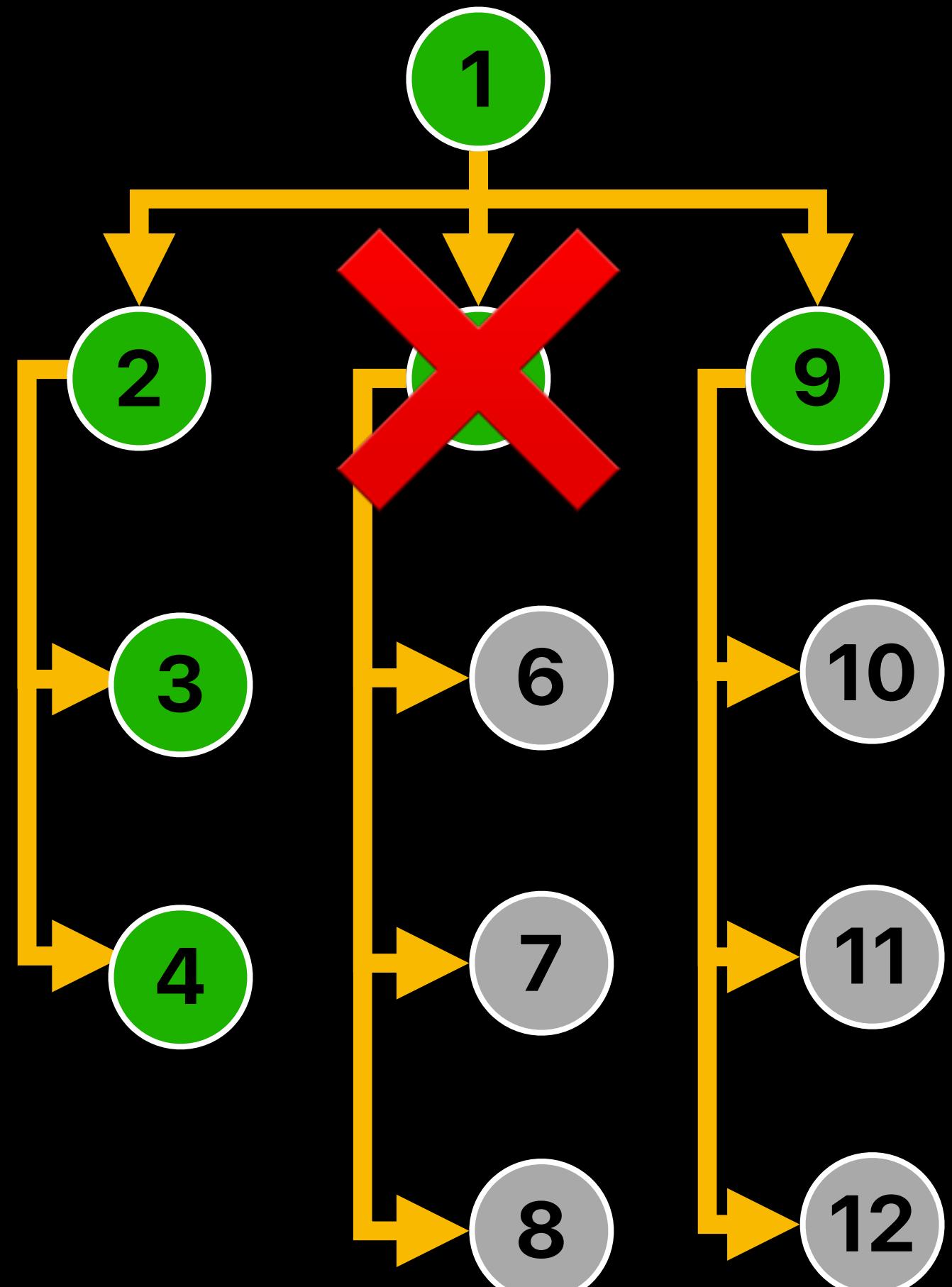
# Track live objects



i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

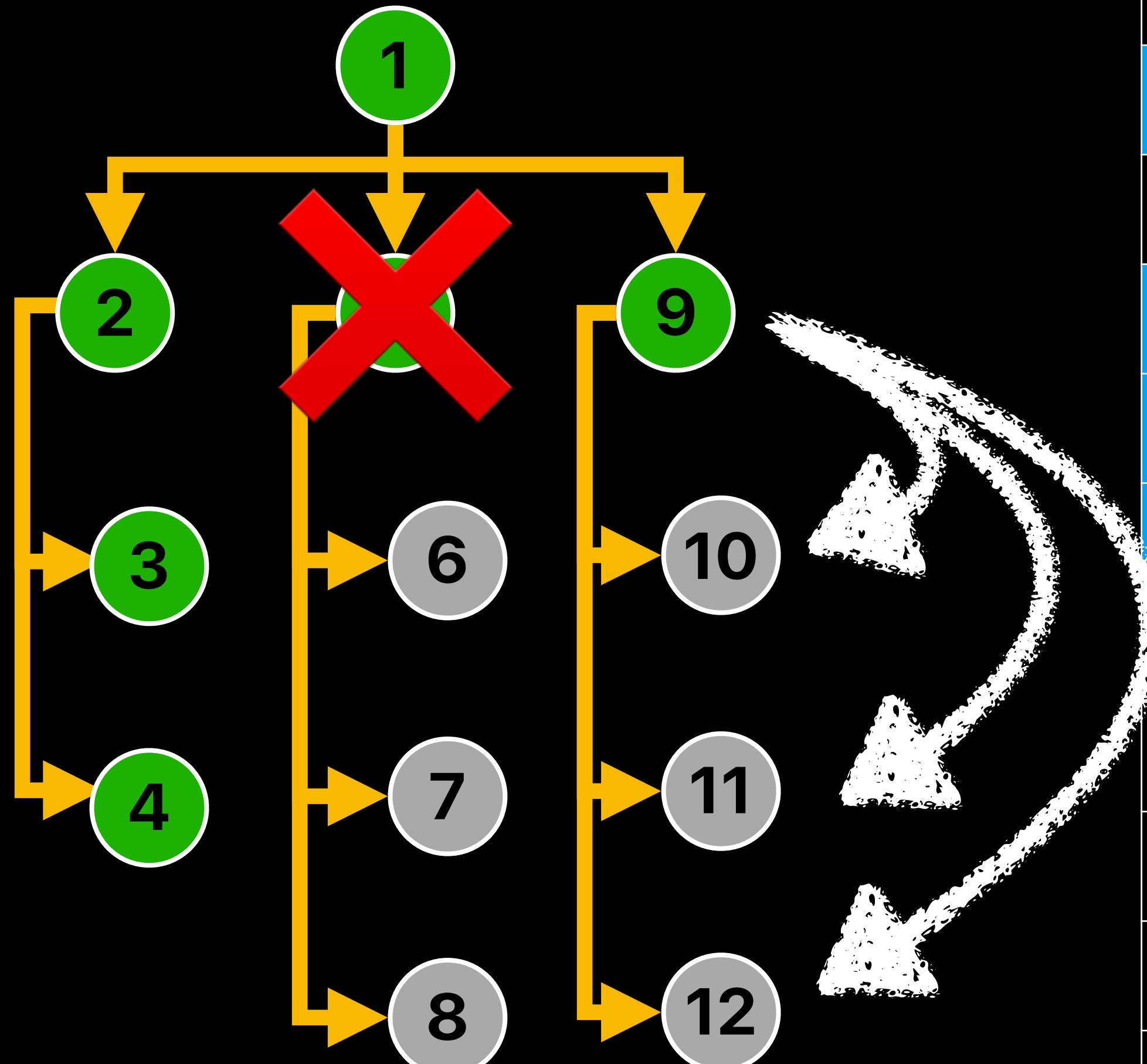


# Track live objects



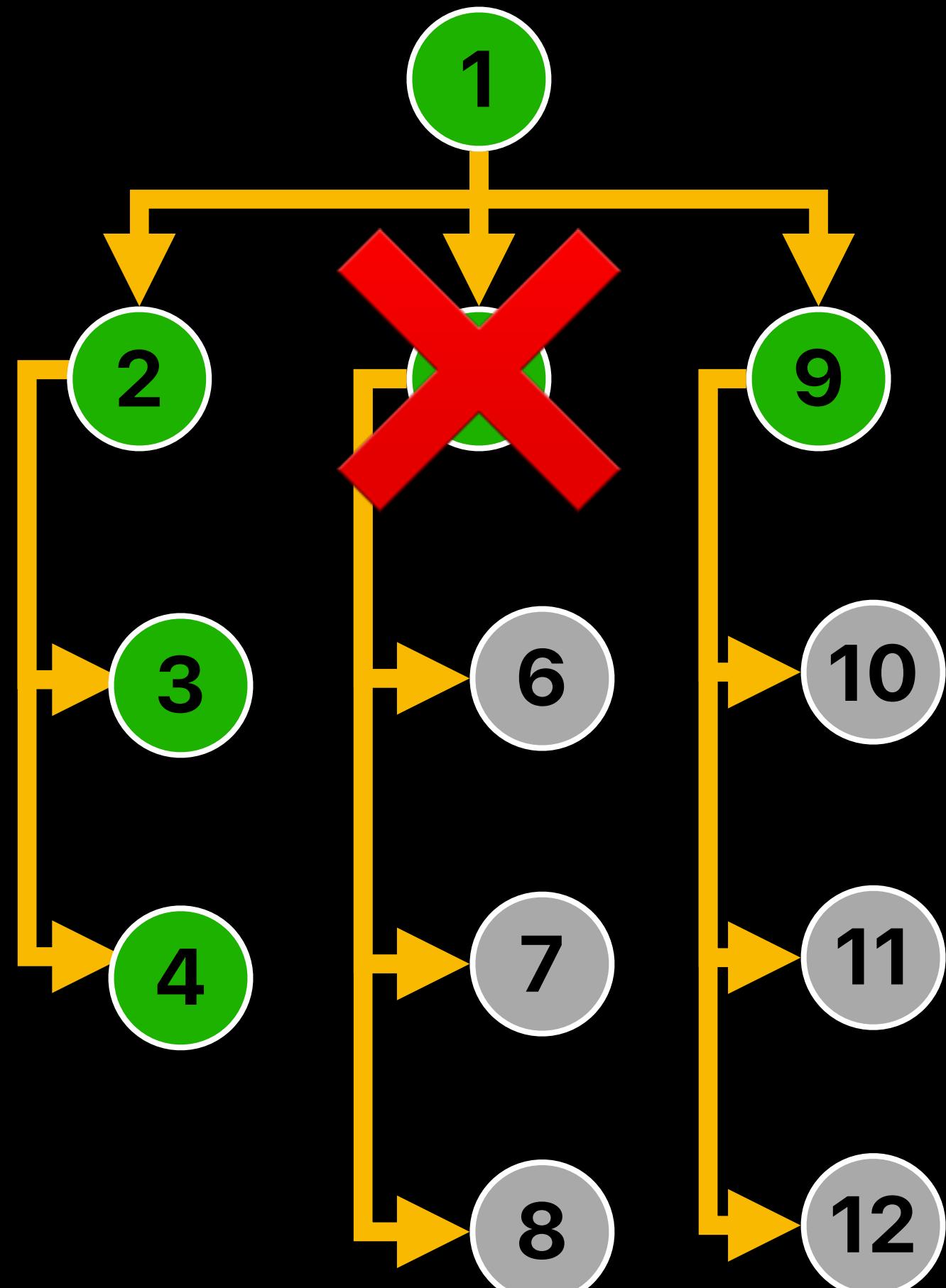
i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



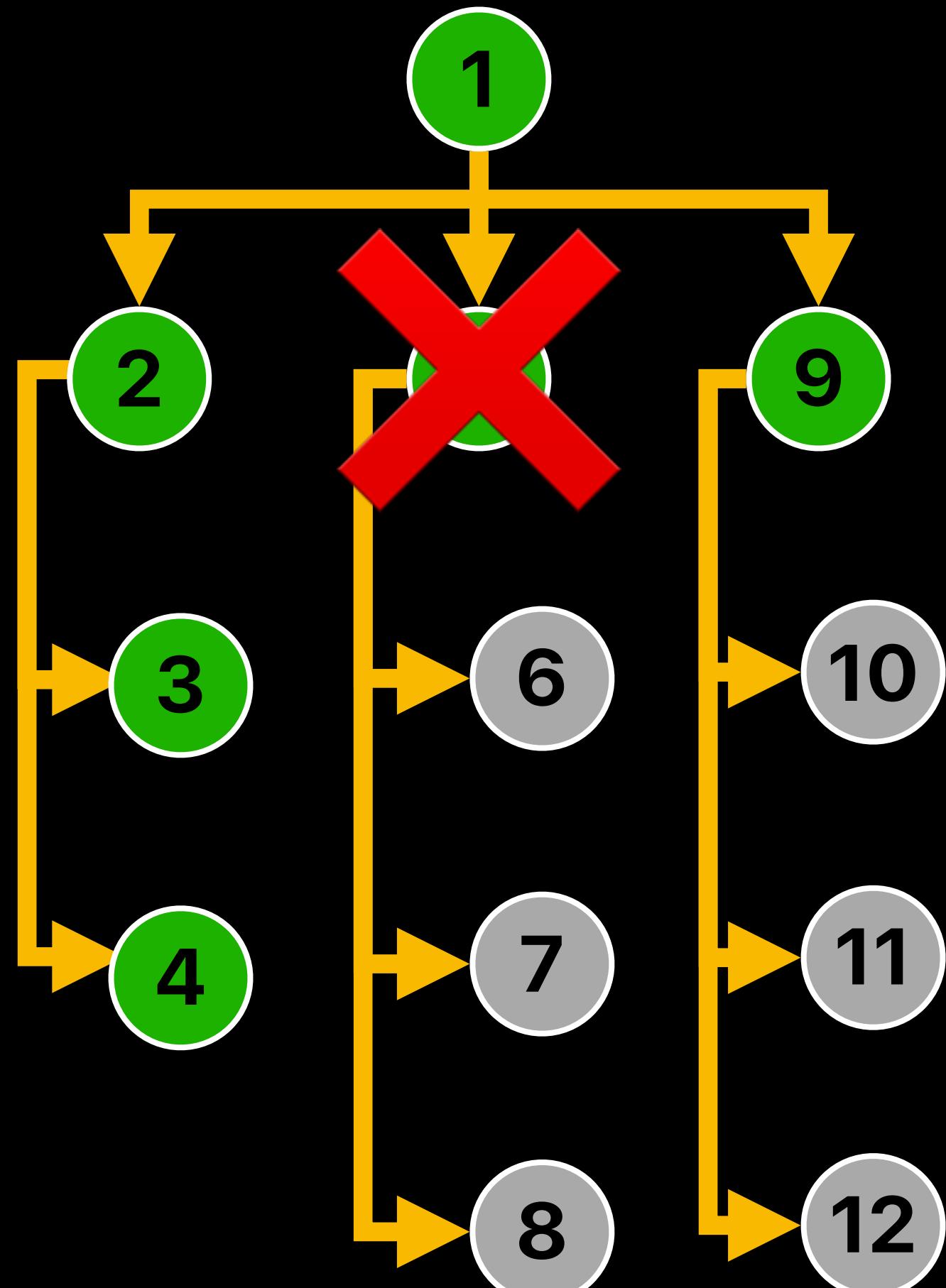
i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



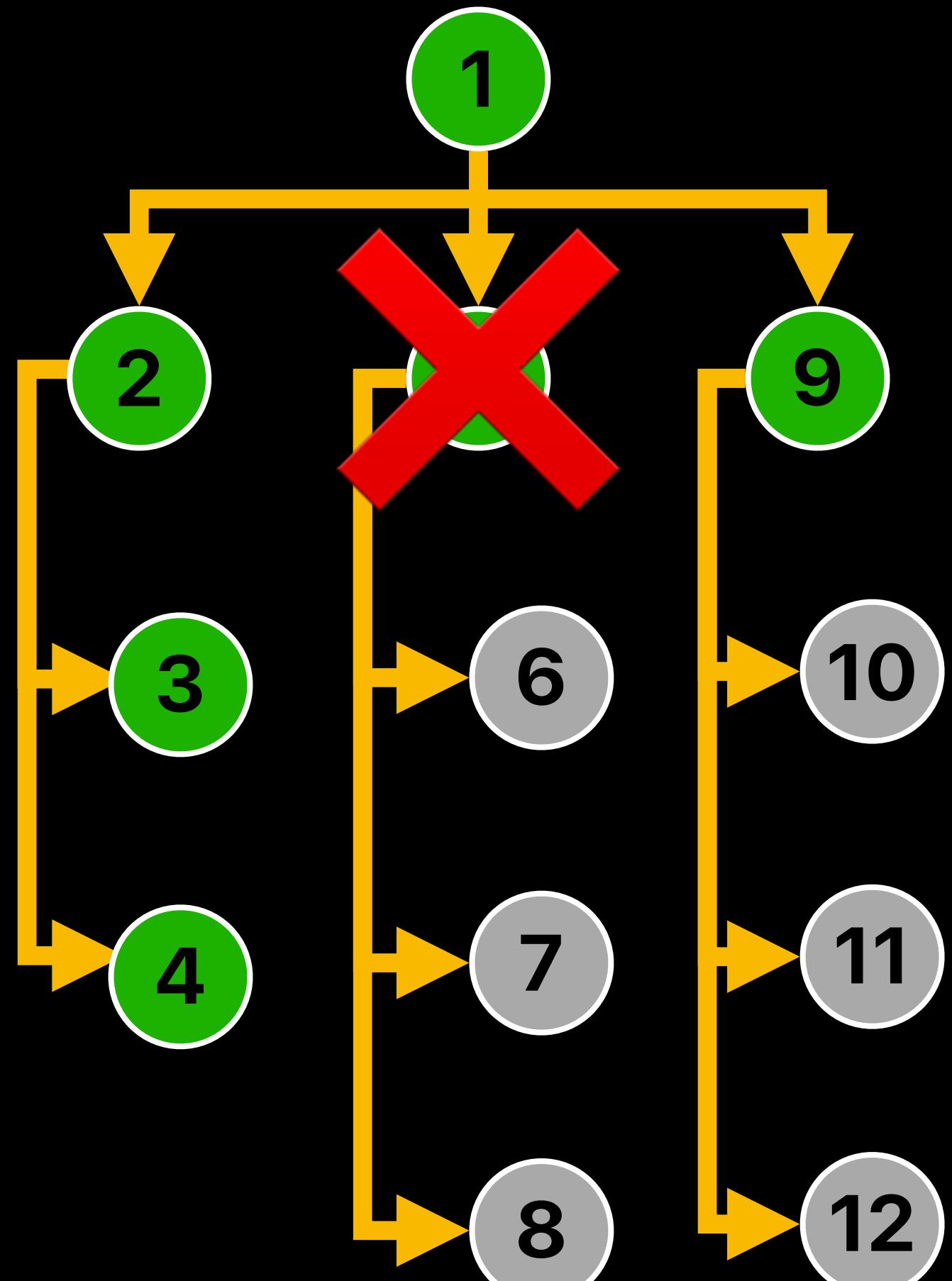
i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



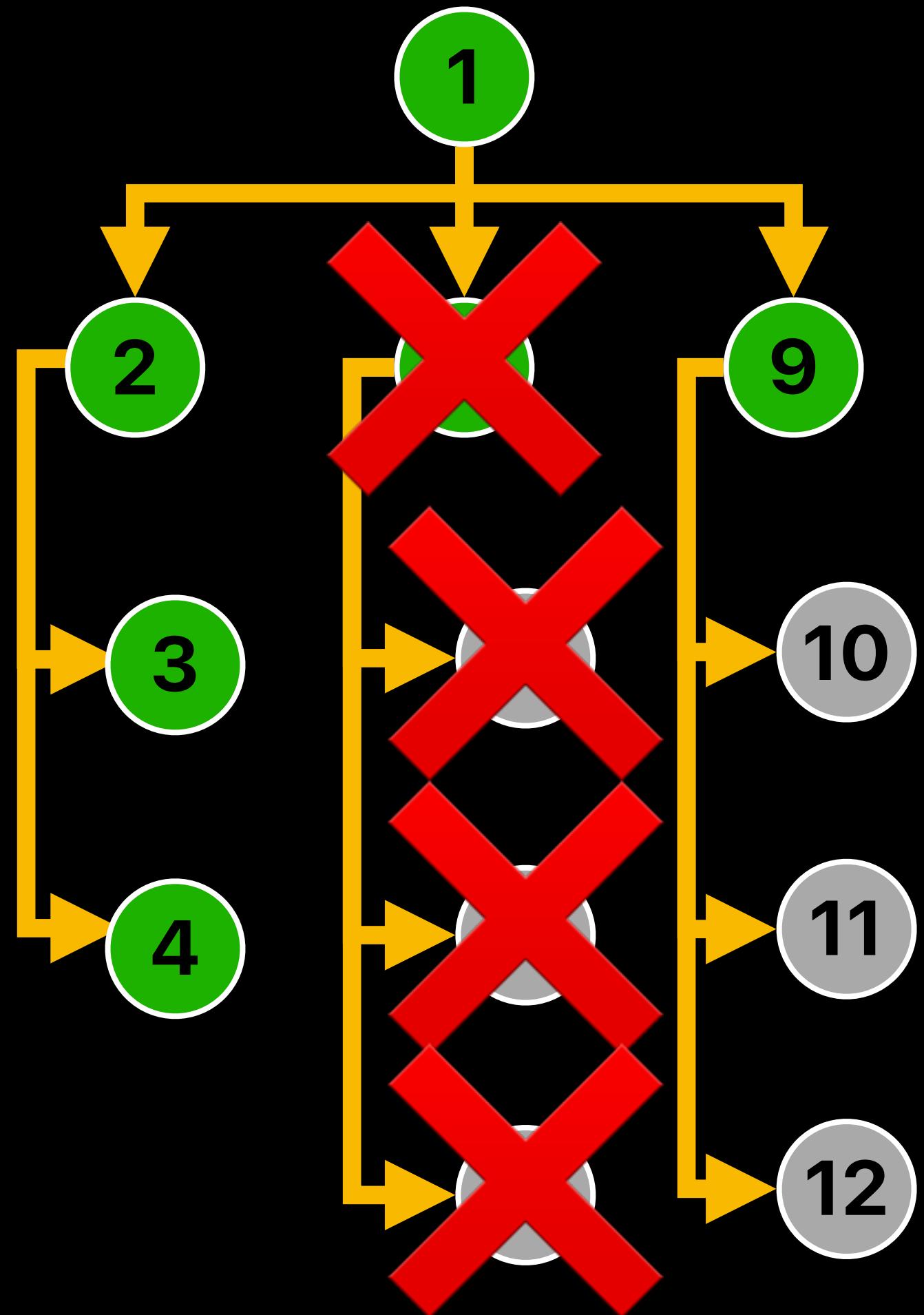
i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	41
9	11	41
9	12	41

# Track live objects



i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	42
9	11	42
9	12	42

# Track live objects



i1	i2	gen
1	2	42
1	5	41
1	9	42
2	3	42
2	4	42
5	6	41
5	7	41
5	8	41
9	10	42
9	11	42
9	12	42

# Track live objects

```
CREATE TABLE ooref (i1 INTEGER, i2 INTEGER, gen INTEGER);
```

# Track live objects

```
WITH RECURSIVE children(i1, i2) AS (
    SELECT i1, i2 FROM ooref WHERE i1 = ?1 AND gen != ?2
    UNION
    SELECT ooref.i1, ooref.i2 FROM ooref JOIN
        children ON ooref.i1 = children.i2 AND ooref.gen != ?2
)
UPDATE ooref SET gen = ?2
FROM children
WHERE ooref.i1 = children.i1 AND ooref.i2 = children.i2;
```

# Track live objects

```
WITH RECURSIVE children(i1, i2) AS (
    SELECT i1, i2 FROM ooref WHERE i1 = ?1 AND gen != ?2
    UNION
    SELECT ooref.i1, ooref.i2 FROM ooref JOIN
        children ON ooref.i1 = children.i2 AND ooref.gen != ?2
)
UPDATE ooref SET gen = ?2
FROM children
WHERE ooref.i1 = children.i1 AND ooref.i2 = children.i2;
```

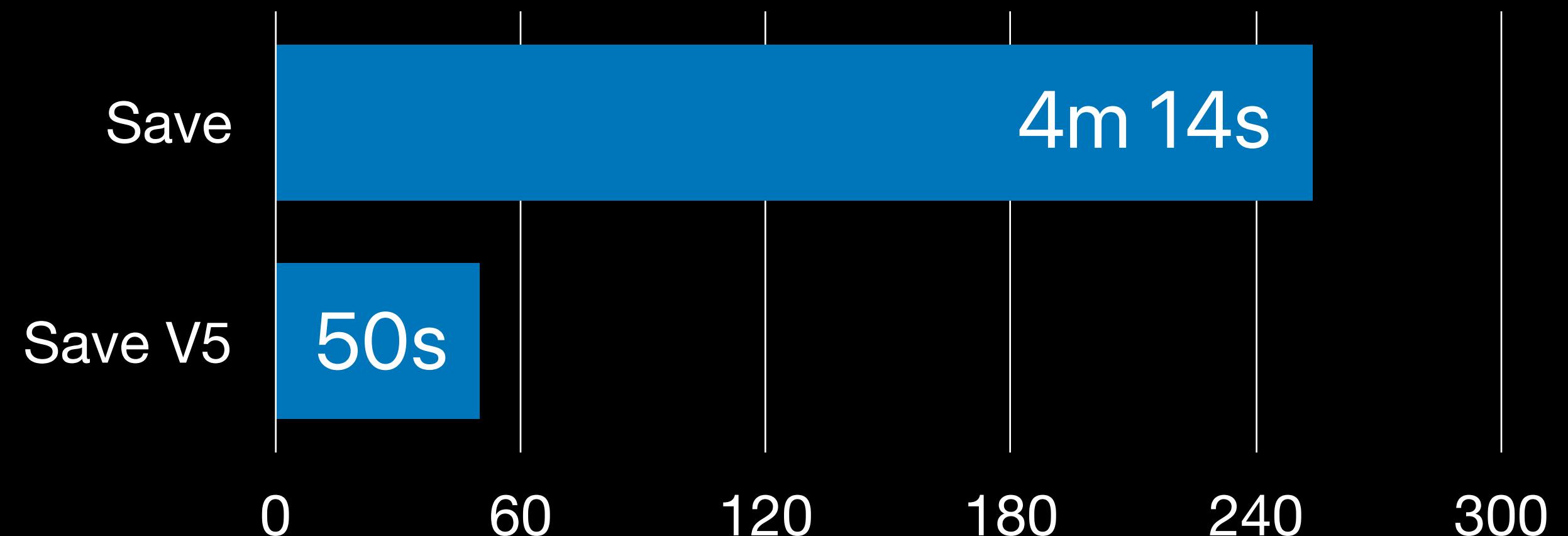
Was it worth it?

# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

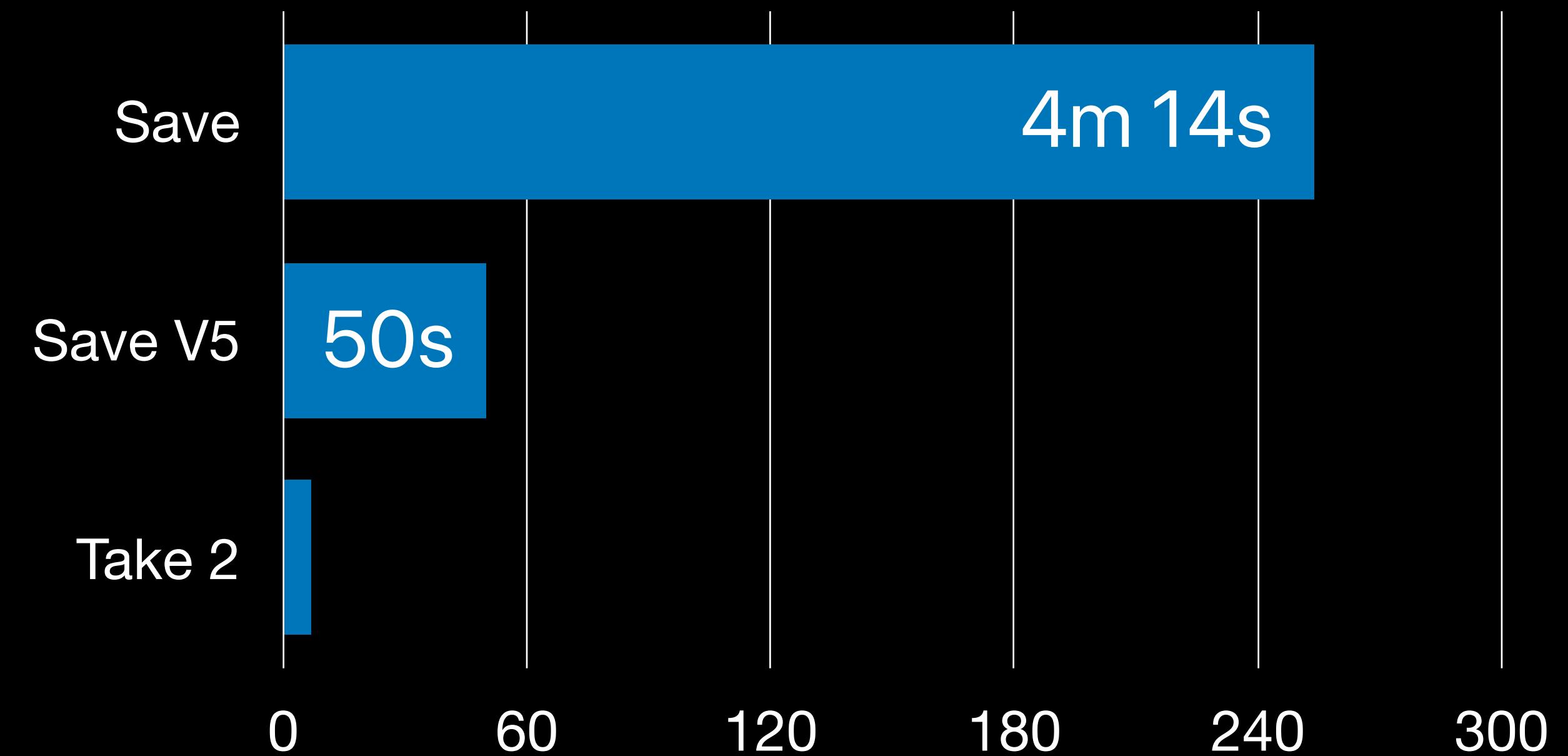


# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

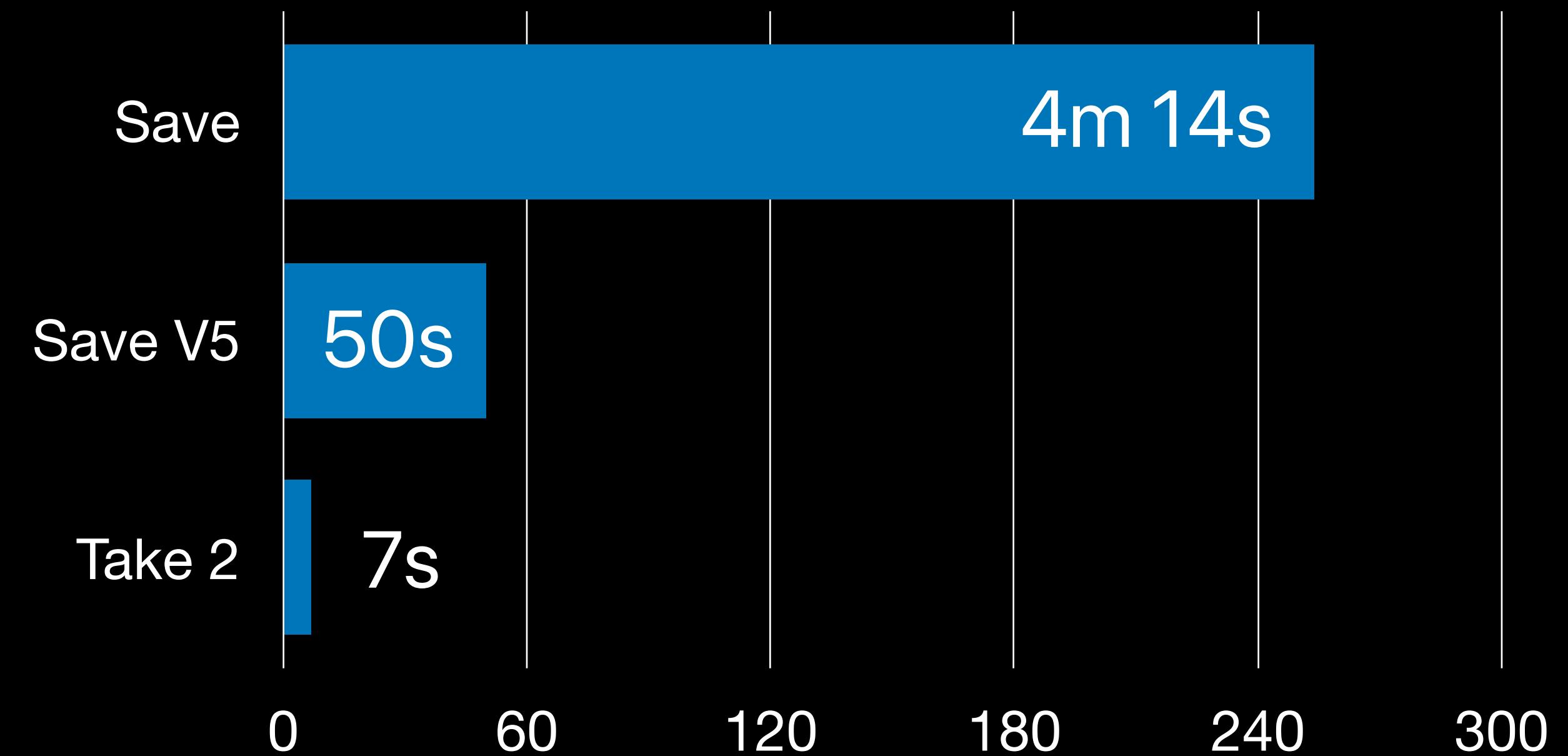


# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files

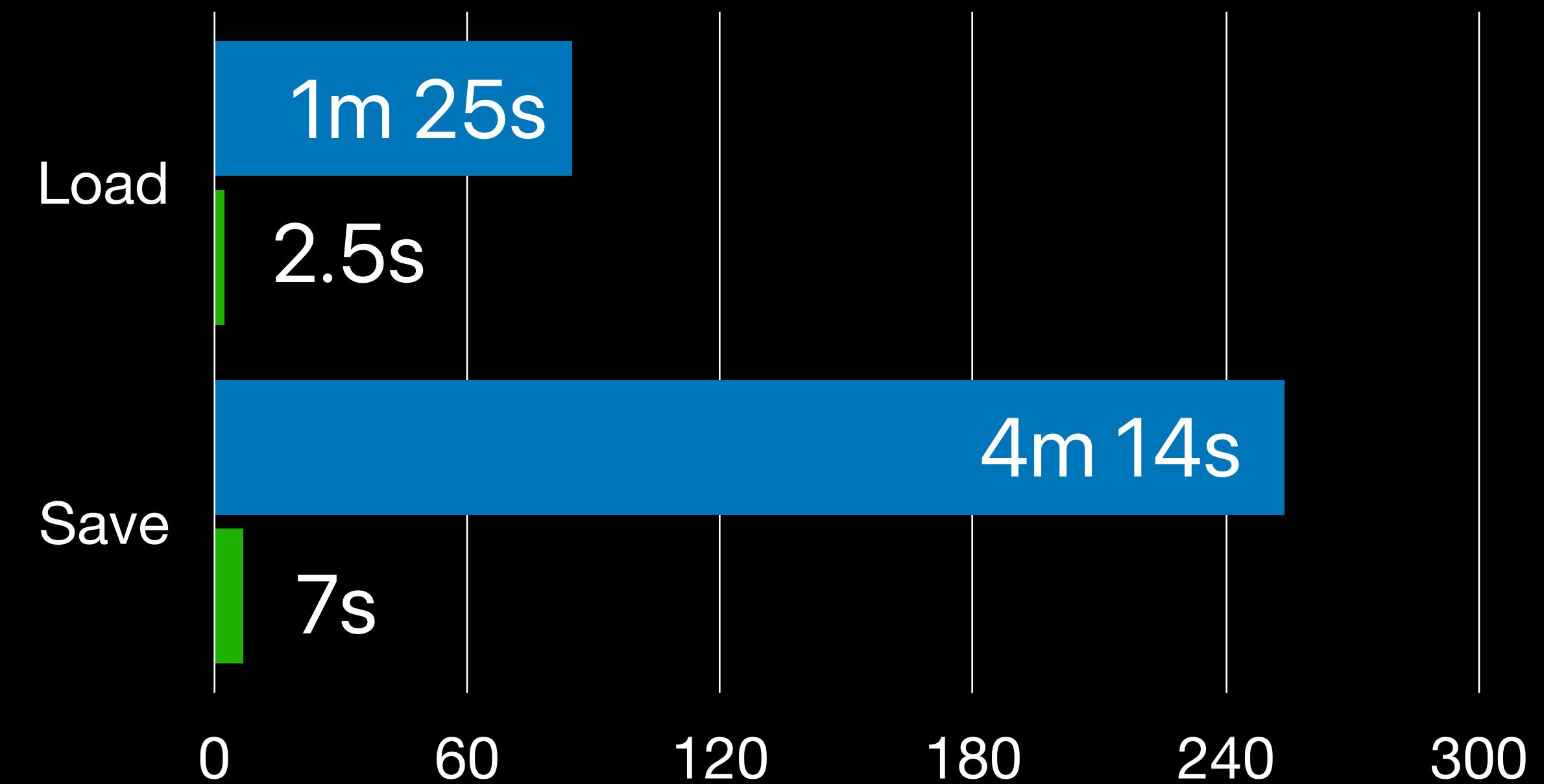


# Improving save time



large website.sparkle

- 35 GB
- 300 pages
- 100k words
- 800 images
- 230 videos
- 400 attached files





3600%

# Thank You!



Me giving  
programming advice

My code