WRITE TESTS LIKE A MATHEMATICIAN

ISAAC J. LEE

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

Running:

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

1 assertion of 1 passed, 0 failed.

If, if, if

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

2 assertions of 2 passed, 0 failed.

- If, if, if
- Use common, everyday words

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

3 assertions of 3 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

4 assertions of 4 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms
- All your basis are belong to us

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

5 assertions of 5 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms
- All your basis are belong to us
- 1 picture = 1000 words

Therefore we will now give a few examples of matrices (represented by their sparsity pattern) and their graphs.

The graph of a matrix often reveals at a single glance its structure, which might not be evident from the sparsity pattern. Thus, consider the matrix

At a first glance, there is nothing to link it to any of the four matrices that we have just displayed, but its graph

tells a different story – it is nothing other than the cyclic matrix in disguise! To see this, just relabel the vertices as follows:

$$1 \to 1, \quad 2 \to 5, \quad 3 \to 2, \quad 4 \to 4, \quad 5 \to 6, \quad 6 \to 3.$$

This, of course, is equivalent to reordering (simultaneously) the equations and variables.

An ordered set of edges $\{(i_k, j_k)\}_{k=1}^{\nu} \subseteq \mathbb{E}$ is called a *path* joining the vertices α and β if $\alpha \in \{i_1, j_1\}$, $\beta \in \{i_\nu, j_\nu\}$ and for every $k = 1, 2, \ldots, \nu - 1$ the set $\{i_k, j_k\} \cap \{i_{k+1}, j_{k+1}\}$ contains exactly one member. It is a *simple path* if it does not visit any vertex more than once. We say that $\mathbb{G}$ is a *tree* if each two members of $\mathbb{V}$ are joined by a unique simple path. Both tridiagonal and arrowhead matrices correspond to trees, but this is not the case with either quindiagonal or cyclic matrices when $\nu \geq 3$.

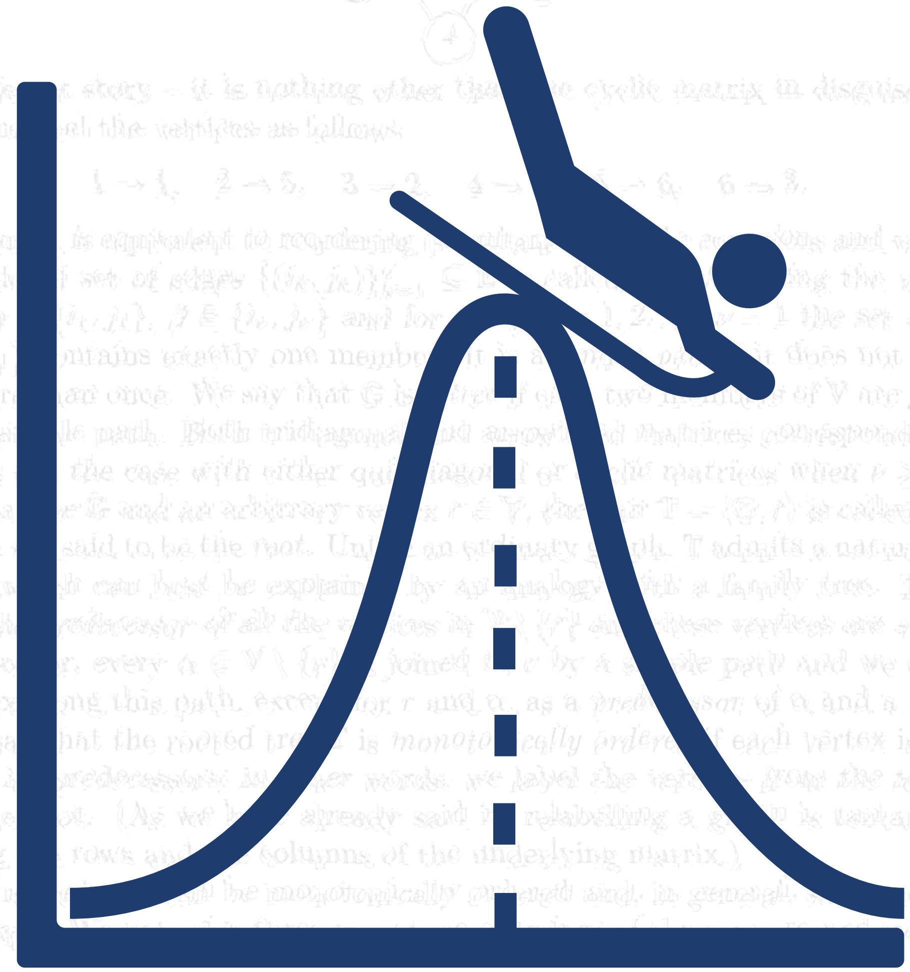
Given a tree $\mathbb{G}$ and an arbitrary vertex $r \in \mathbb{V}$, the pair $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is called a *rooted tree*, while r is said to be the *root*. Unlike an ordinary graph, $\mathbb{T}$ admits a natural partial ordering, which can best be explained by an analogy with a family tree. Thus, the root r is the *predecessor* of all the vertices in $\mathbb{V} \setminus \{r\}$ and these vertices are *successors* of r. Moreover, every $\alpha \in \mathbb{V} \setminus \{r\}$ is joined to r by a simple path and we designate each vertex along this path, except for r and α, as a *predecessor* of α and a *successor* of r. We say that the rooted tree $\mathbb{T}$ is *monotonically ordered* if each vertex is labelled before all its predecessors; in other words, we label the vertices from the top of the tree to the root. (As we have already said it, relabelling a graph is tantamount to permuting the rows and the columns of the underlying matrix.)

Every rooted tree can be monotonically ordered and, in general, such an ordering is not unique. We now give three monotone orderings of the same rooted tree:

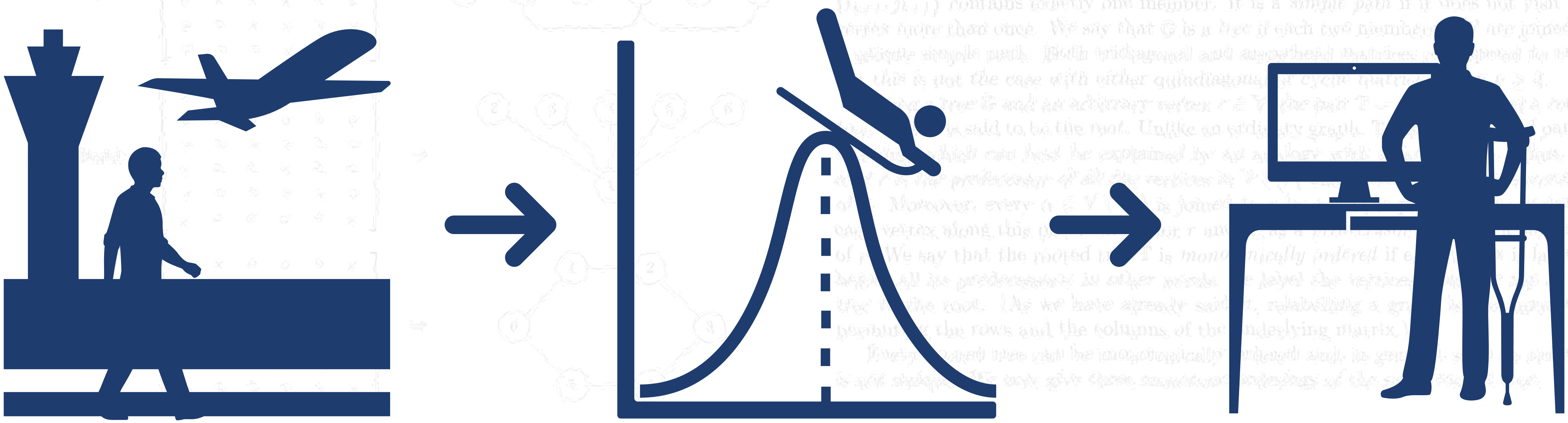
Theorem 11.1 *Let A be a symmetric matrix whose graph $\mathbb{G}$ is a tree. Choose a root $r \in \{1, 2, \ldots, d\}$ and assume that the rows and columns of A have been arranged so that $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is monotonically ordered. Given that $A = LL^\top$ is a Cholesky factorization, it is true that*

$$\ell_{k,j} = \frac{a_{k,j}}{\ell_{j,j}}, \qquad k = j+1, j+2, \ldots, d, \quad j = 1, 2, \ldots, d-1. \qquad (11.5)$$

IF



THEN



```
assert.speaker().getsPersonal();

await sing('Happy Birthday');

assert.audience().isHappy();
```

SUSPECT 1

OBSERVER / COMPUTED

SUSPECT 2

EMBER DATA / FORM BUILDER

Therefore we will now give a few examples of matrices (represented by their sparsity patterns) and their graphs.

tridiagonal: $\begin{bmatrix} \times & \times & \circ & \circ & \circ & \circ \\ \times & \times & \times & \circ & \circ & \circ \\ \circ & \times & \times & \times & \circ & \circ \\ \circ & \circ & \times & \times & \times & \circ \\ \circ & \circ & \circ & \times & \times & \times \\ \circ & \circ & \circ & \circ & \times & \times \end{bmatrix} \Rightarrow$ 

quindiagonal: $\begin{bmatrix} \times & \times & \times & \circ & \circ & \circ \\ \times & \times & \times & \times & \circ & \circ \\ \times & \times & \times & \times & \times & \circ \\ \circ & \times & \times & \times & \times & \times \\ \circ & \circ & \times & \times & \times & \times \\ \circ & \circ & \circ & \times & \times & \times \end{bmatrix} \Rightarrow$

arrowhead: $\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \circ & \circ & \circ & \circ \\ \times & \circ & \times & \circ & \circ & \circ \\ \times & \circ & \circ & \times & \circ & \circ \\ \times & \circ & \circ & \circ & \times & \circ \\ \times & \circ & \circ & \circ & \circ & \times \end{bmatrix} \Rightarrow$

cyclic: $\begin{bmatrix} \times & \times & \circ & \circ & \circ & \times \\ \times & \times & \times & \circ & \circ & \circ \\ \circ & \times & \times & \times & \circ & \circ \\ \circ & \circ & \times & \times & \times & \circ \\ \circ & \circ & \circ & \times & \times & \times \\ \times & \circ & \circ & \circ & \times & \times \end{bmatrix} \Rightarrow$ 

The graph of a matrix often reveals at a single glance its structure, which might not be evident from the sparsity pattern. Thus, consider the matrix

$$\begin{bmatrix} \times & \circ & \times & \circ & \times & \circ \\ \circ & \times & \circ & \times & \times & \circ \\ \times & \circ & \times & \circ & \circ & \times \\ \circ & \times & \circ & \times & \circ & \times \\ \times & \times & \circ & \circ & \times & \circ \\ \circ & \circ & \times & \times & \circ & \times \end{bmatrix}.$$

At a first glance, there is nothing to link it to any of the four matrices that we have just displayed, but its graph,

tells a different story – it is nothing other than the cyclic matrix in disguise! To see this, just relabel the vertices as follows

$$1 \to 1, \quad 2 \to 5, \quad 3 \to 2, \quad 4 \to 4, \quad 5 \to 6, \quad 6 \to 3.$$

This, of course, is equivalent to reordering (simultaneously) the equations and variables.

An ordered set of edges $\{(i_k, j_k)\}_{k=1}^{\nu} \subseteq \mathbb{E}$ is called a *path* joining the vertices α and β if $\alpha \in \{i_1, j_1\}$, $\beta \in \{i_\nu, j_\nu\}$ and for every $k = 1, 2, \ldots, \nu - 1$ the set $\{i_k, j_k\} \cap \{i_{k+1}, j_{k+1}\}$ contains exactly one member. It is a *simple path* if it does not visit any vertex more than once. We say that $\mathbb{G}$ is a *tree* if each two members of $\mathbb{V}$ are joined by a unique simple path. Both tridiagonal and arrowhead matrices correspond to trees, but this is not the case with either quindiagonal or cyclic matrices when $\nu \geq 3$.

Given a tree $\mathbb{G}$ and an arbitrary vertex $r \in \mathbb{V}$, the pair $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is called a *rooted tree*, while r is said to be the *root*. Unlike an ordinary graph, $\mathbb{T}$ admits a natural partial ordering, which can best be explained by an analogy with a family tree. Thus, the root r is the *predecessor* of all the vertices in $\mathbb{V} \setminus \{r\}$ and these vertices are *successors* of r. Moreover, every $\alpha \in \mathbb{V} \setminus \{r\}$ is joined to r by a simple path and we designate each vertex along this path, except for r and α, as a *predecessor* of α and a *successor* of r. We say that the rooted tree $\mathbb{T}$ is *monotonically ordered* if each vertex is labelled before all its predecessors; in other words, we label the vertices from the top of the tree to the root. (As we have already said it, relabelling a graph is tantamount to permuting the rows and the columns of the underlying matrix.)

Every rooted tree can be monotonically ordered and, in general, such an ordering is not unique. We now give three monotone orderings of the same rooted tree:

Theorem 11.1 *Let A be a symmetric matrix whose graph $\mathbb{G}$ is a tree. Choose a root $r \in \{1, 2, \ldots, d\}$ and assume that the rows and columns of A have been arranged so that $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is monotonically ordered. Given that $A = LL^\top$ is a Cholesky factorization, it is true that*

$$\ell_{k,j} = \frac{a_{k,j}}{\ell_{j,j}}, \qquad k = j+1, j+2, \ldots, d, \quad j = 1, 2, \ldots, d-1. \tag{11.5}$$

Therefore we will now give a few examples of matrices (represented by their sparsity pattern) and their graphs.

tridiagonal: $\begin{bmatrix} \times & \times & \circ & \circ & \circ & \circ \\ \times & \times & \times & \circ & \circ & \circ \\ \circ & \times & \times & \times & \circ & \circ \\ \circ & \circ & \times & \times & \times & \circ \\ \circ & \circ & \circ & \times & \times & \times \\ \circ & \circ & \circ & \circ & \times & \times \end{bmatrix} \Rightarrow$ 

quindiagonal: $\begin{bmatrix} \times & \times & \times & \circ & \circ & \circ \\ \times & \times & \times & \times & \circ & \circ \\ \times & \times & \times & \times & \times & \circ \\ \circ & \times & \times & \times & \times & \times \\ \circ & \circ & \times & \times & \times & \times \\ \circ & \circ & \circ & \times & \times & \times \end{bmatrix} \Rightarrow$

arrowhead: $\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \circ & \circ & \circ & \circ \\ \times & \circ & \times & \circ & \circ & \circ \\ \times & \circ & \circ & \times & \circ & \circ \\ \times & \circ & \circ & \circ & \times & \circ \\ \times & \circ & \circ & \circ & \circ & \times \end{bmatrix} \Rightarrow$

cyclic: $\begin{bmatrix} \times & \times & \circ & \circ & \circ & \times \\ \times & \times & \times & \circ & \circ & \circ \\ \circ & \times & \times & \times & \circ & \circ \\ \circ & \circ & \times & \times & \times & \circ \\ \circ & \circ & \circ & \times & \times & \times \\ \times & \circ & \circ & \circ & \times & \times \end{bmatrix} \Rightarrow$

The graph of a matrix often reveals at a single glance its structure, which might not be evident from the sparsity pattern. Thus, consider the matrix

$$\begin{bmatrix} \times & \circ & \times & \circ & \times & \circ \\ \circ & \times & \circ & \times & \times & \circ \\ \times & \circ & \times & \circ & \circ & \times \\ \circ & \times & \circ & \times & \circ & \times \\ \times & \times & \circ & \circ & \times & \circ \\ \circ & \circ & \times & \times & \circ & \times \end{bmatrix}.$$

At a first glance, there is nothing to link it to any of the four matrices that we have just displayed, but its graph,

tells a different story – it is nothing other than the cyclic matrix in disguise. To see this, just relabel the vertices as follows:

$$1 \to 1, \quad 2 \to 5, \quad 3 \to 2, \quad 4 \to 4, \quad 5 \to 6, \quad 6 \to 3.$$

This, of course, is equivalent to reordering (simultaneously) the equations and variables.

An ordered set of edges $\{(i_k, j_k)\}_{k=1}^{\nu} \subseteq \mathbb{E}$ is called a *path* joining the vertices α and β if $\alpha \in \{i_1, j_1\}$, $\beta \in \{i_\nu, j_\nu\}$ and for every $k = 1, 2, \ldots, \nu - 1$ the set $\{i_k, j_k\} \cap \{i_{k+1}, j_{k+1}\}$ contains exactly one member. It is a *simple path* if it does not visit any vertex more than once. We say that $\mathbb{G}$ is a *tree* if each two members of $\mathbb{V}$ are joined by a unique simple path. Both tridiagonal and arrowhead matrices correspond to trees, but this is not the case with either quindiagonal or cyclic matrices when $\nu \geq 3$.

Given a tree $\mathbb{G}$ and an arbitrary vertex $r \in \mathbb{V}$, the pair $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is called a *rooted tree*, while r is said to be the *root*. Unlike an ordinary graph, $\mathbb{T}$ admits a natural partial ordering, which can best be explained by an analogy with a family tree. Thus, the root r is the *predecessor* of all the vertices in $\mathbb{V} \setminus \{r\}$ and these vertices are *successors* of r. Moreover, every $\alpha \in \mathbb{V} \setminus \{r\}$ is joined to r by a simple path and we designate each vertex along this path, except for r and α, as a *predecessor* of α and a *successor* of r. We say that the rooted tree $\mathbb{T}$ is *monotonically ordered* if each vertex is labelled before all its predecessors; in other words, we label the vertices from the top of the tree to the root. (As we have already said it, relabelling a graph is tantamount to permuting the rows and the columns of the underlying matrix.)

Every rooted tree can be monotonically ordered and, in general, such an ordering is not unique. We now give three monotone orderings of the same rooted tree:

Theorem 11.1 *Let A be a symmetric matrix whose graph $\mathbb{G}$ is a tree. Choose a root $r \in \{1, 2, \ldots, d\}$ and assume that the rows and columns of A have been arranged so that $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is monotonically ordered. Given that $A = LL^\top$ is a Cholesky factorization, it is true that*

$$\ell_{k,j} = \frac{a_{k,j}}{\ell_{j,j}}, \qquad k = j+1, j+2, \ldots, d, \quad j = 1, 2, \ldots, d-1. \tag{11.5}$$

SUSPECT 5

ADMIN PRIVILEGE

2

USE COMMON, EVERYDAY WORDS

COMMON

CONVENTION;

AGREED ON BY MANY

EVERYDAY

SIMPLE;

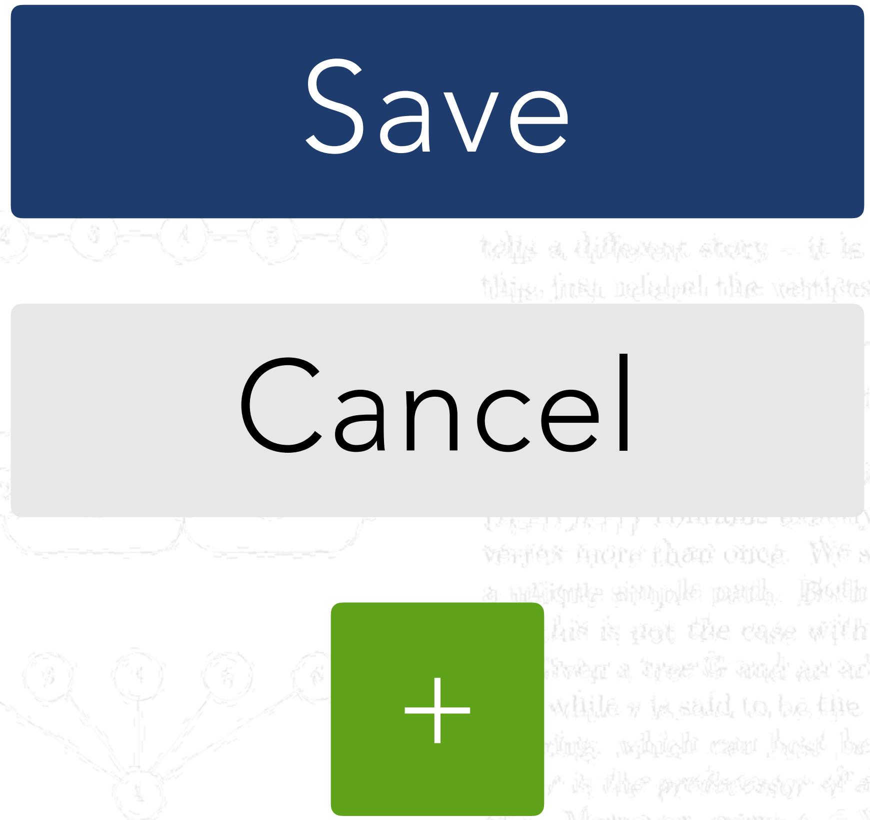
FAMILIAR TO MANY

IF f IS CONTINUOUS IN $[a, b]$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).

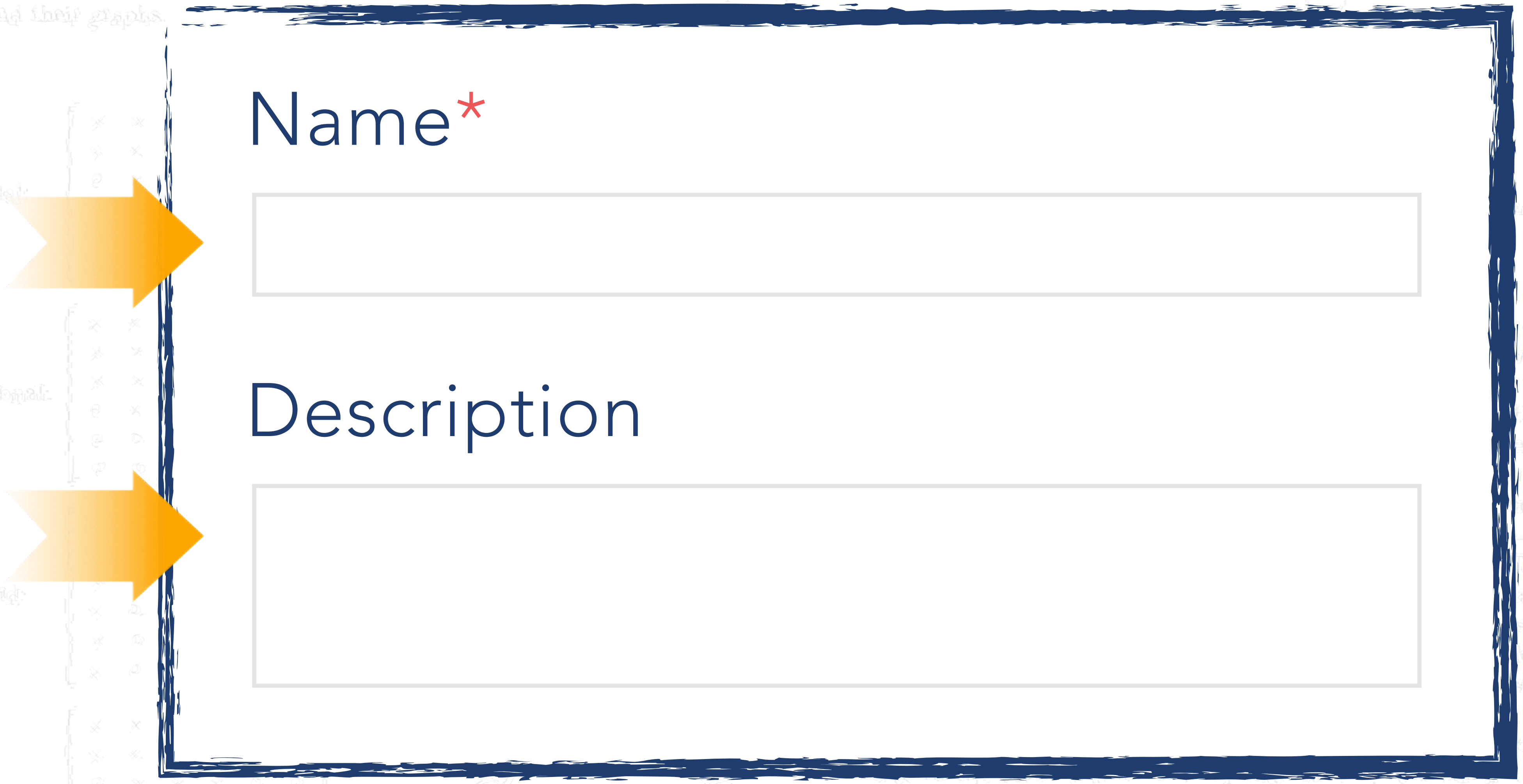
```
assert.dom('[data-test-message]')
  .hasText(
    'Thanks for signing up!',
    'The user sees a welcome message.'
  );
```

IF f IS CONTINUOUS IN $[a, b]$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).


```
'[data-test-link="Dashboard"]'
'[data-test-link="Explore"]'
'[data-test-link="Settings"]'
```




```
'[data-test-button="Save"]'
'[data-test-button="Cancel"]'
'[data-test-button="Add item"]'
```




```
'[data-test-field="Name"]'
'[data-test-field="Description"]'
```

Name

Little Bobby Tables

Description

Better not drop me!

```
'[data-test-field="Name"]'
'[data-test-field="Description"]'
```

3

WRITE LESS WITH THEOREMS AND NEW TERMS

IF f IS CONTINUOUS IN $[a, b]$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).

PROOF.

USE THE INTERMEDIATE VALUE THEOREM.

PROOF.

USE IVT.

THEOREM

PROVEN

TO BE TRUE

```
import { fillIn } from '@ember/test-helpers';

export async function fillForm(fields) {
  for (const { label, value } of fields) {
    // input, textarea
    await fillIn(`[data-test-field="${label}"]`, value);
  }
};
```

```
import { fillForm } from '../helpers/my-test-helpers';
...
test('User can create account', async function(assert) {
  await visit('/signup');
  await fillForm([
    { label: 'Name', value: 'Little Bobby Tables' },
    { label: 'Email', value: 'little.bobby@gmail.com' }
  ]);
  ...
});
```

NEW TERM

UBIQUITOUS

IDEA

IF f IS CONTINUOUS IN $[a, b]$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).

IF $f \in C([a, b])$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).

```
hooks.beforeEach(function(assert) {
  ...
  // Example: assert.isEnabled('Submit', 'Woot!');
  assert.isEnabled = (label, message) => {
    assert.dom(`[data-test-button="${label}"]`)
      .doesNotHaveAttribute('disabled', message);
  };
  ...
});
```

4

ALL YOUR BASIS ARE BELONG TO US

BASIS

INDEPENDENT VECTORS

THAT

SPAN THE ENTIRE SPACE

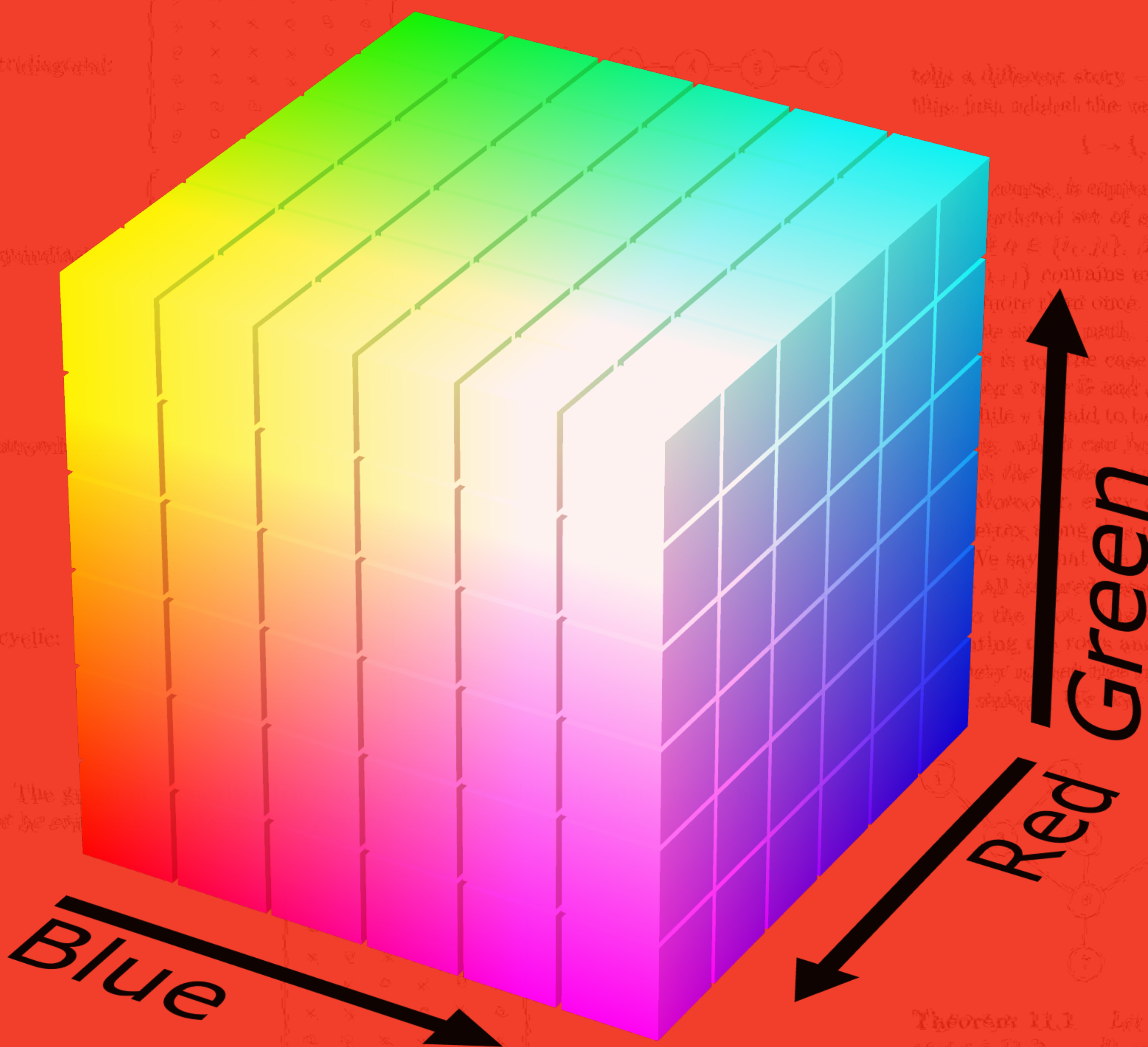
LATITUDE

55.676°

LONGITUDE

12.568°


LATITUDE
30.267°
LONGITUDE
-97.743°



RED
224

GREEN
78

BLUE
57

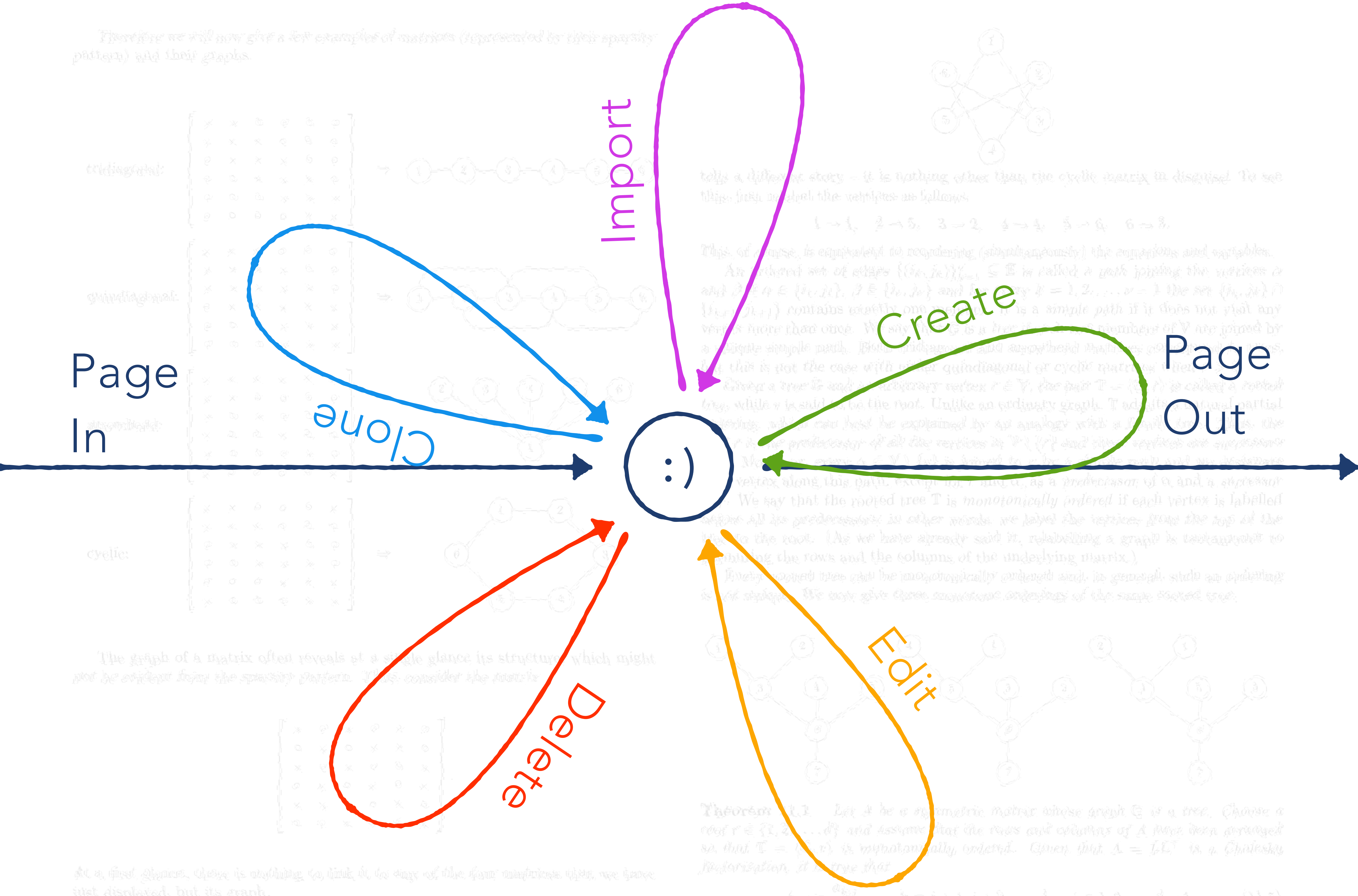
BASIS

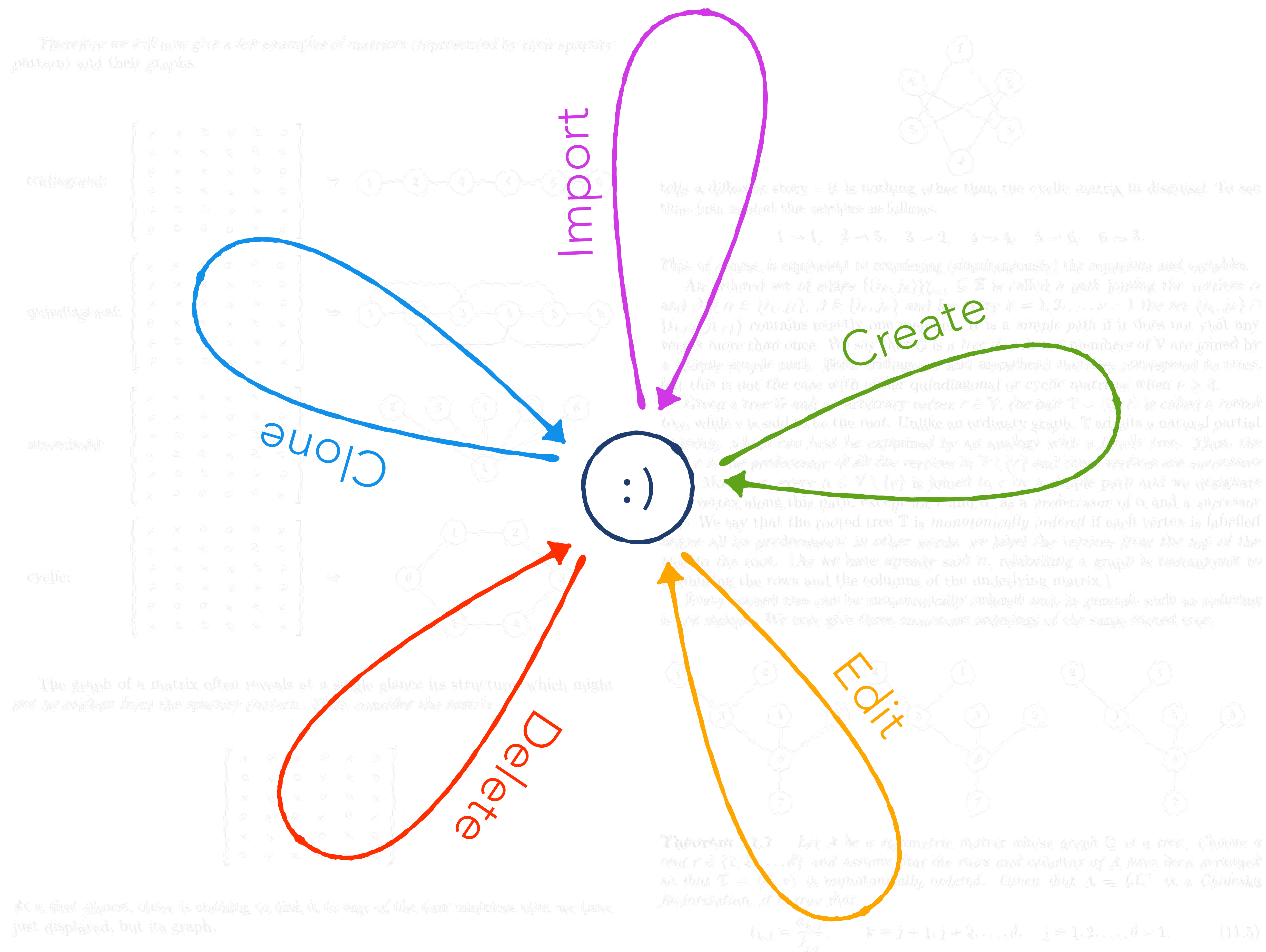
BUILDING BLOCKS

OF TESTS

Create Edit Delete Clone Import

	Name	Description
	Little Bobby Tables	Better not drop me!
	Big Bobby Tables	
	Foo Bar	Making up names is hard.

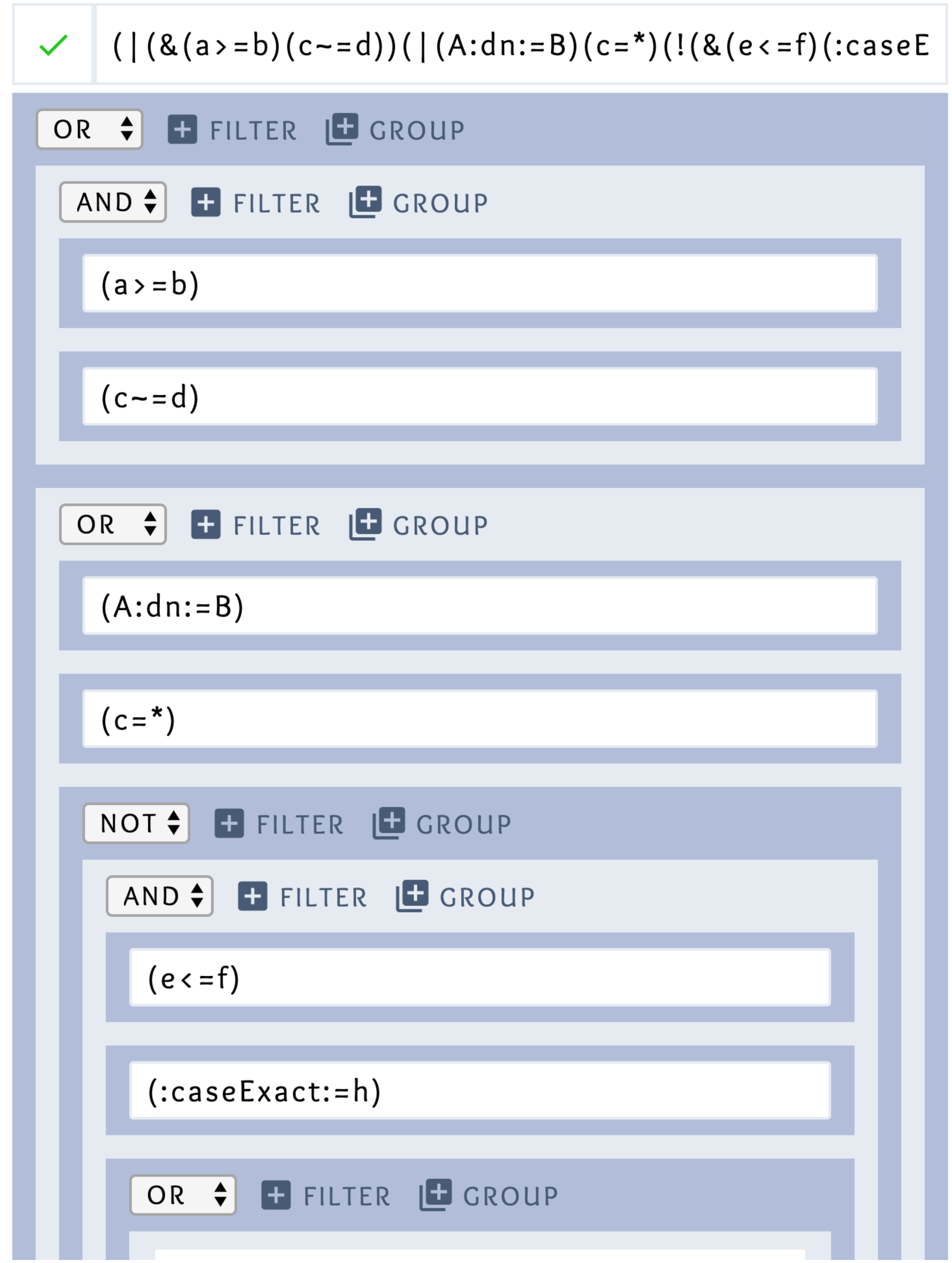

Page In
Clone
Import
:)
Create
Page Out
Delete
Edit



Import
Create
Edit
Delete
Clone
:)


Page
In
:)
Page
Out

```
(|(&(a>=b)(c~=d))
(|(A:dn:=B)(c=*)
(!(&(e<=f)
(:caseExact:=h)
(|(i=j)(!(k<=l)))))))
```



```
(|(!(a>=b)(c~d))
(|(A:dn := B)(c=*)
(!(&(e<=f)
(:caseExact:=h)
(|(i=j)(!(k<=l)))))))
```

(id>=123)

SIMPLE

(email=*)

PRESENT

(cn=Jane*)

SUBSTRING

(:dn:1.2.3.4:=Jay)

EXTENSIBLE

(&(   )...(   ))

AND

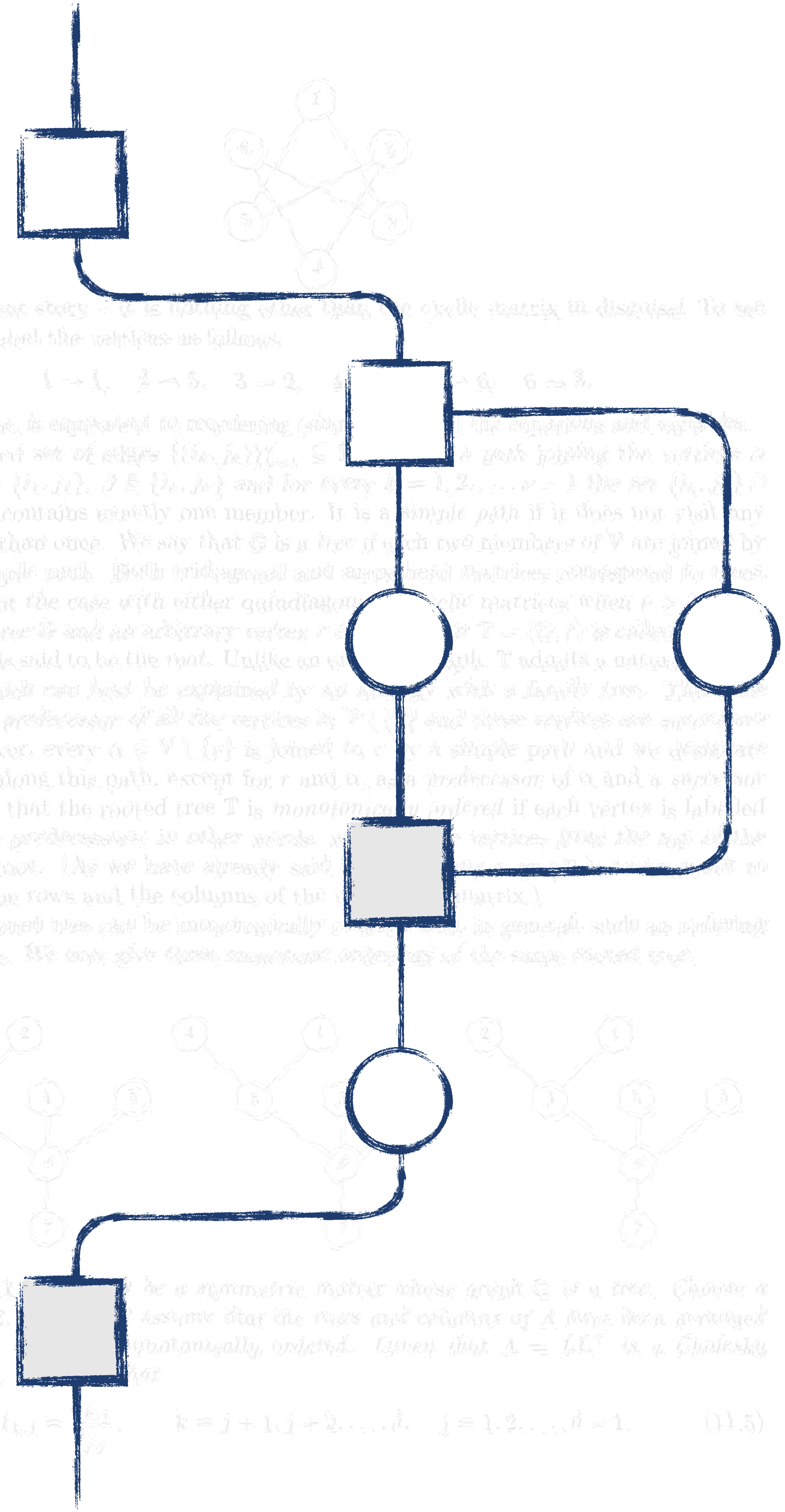
(|(   )...(   ))

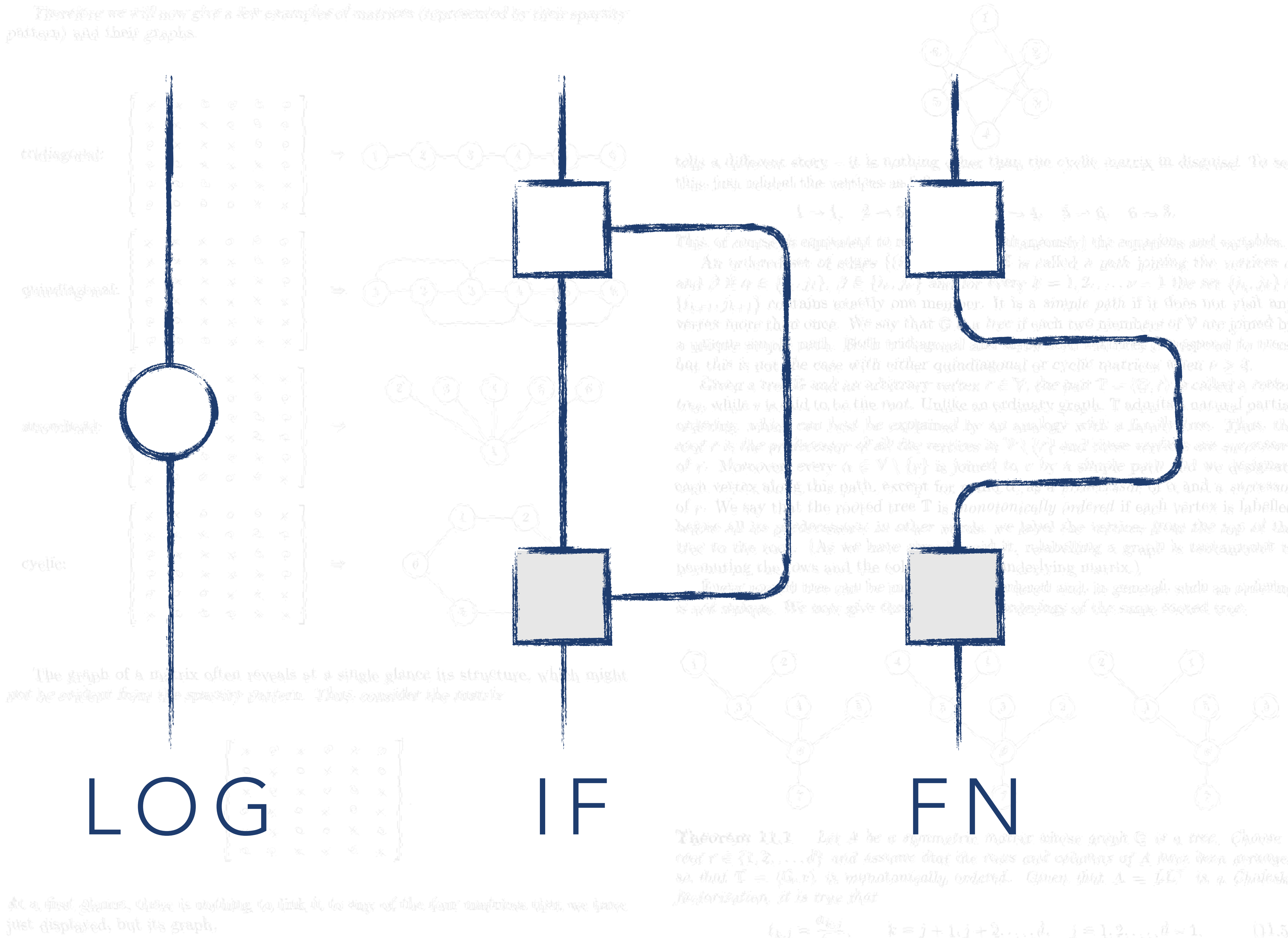
OR

(!(   ))

NOT

```
function addOne(x) {
  if (Number.isFinite(x)) {
    x = x + 1;
  } else {
    console.log('error');
  }
  return x;
}
```




LOG
IF
FN

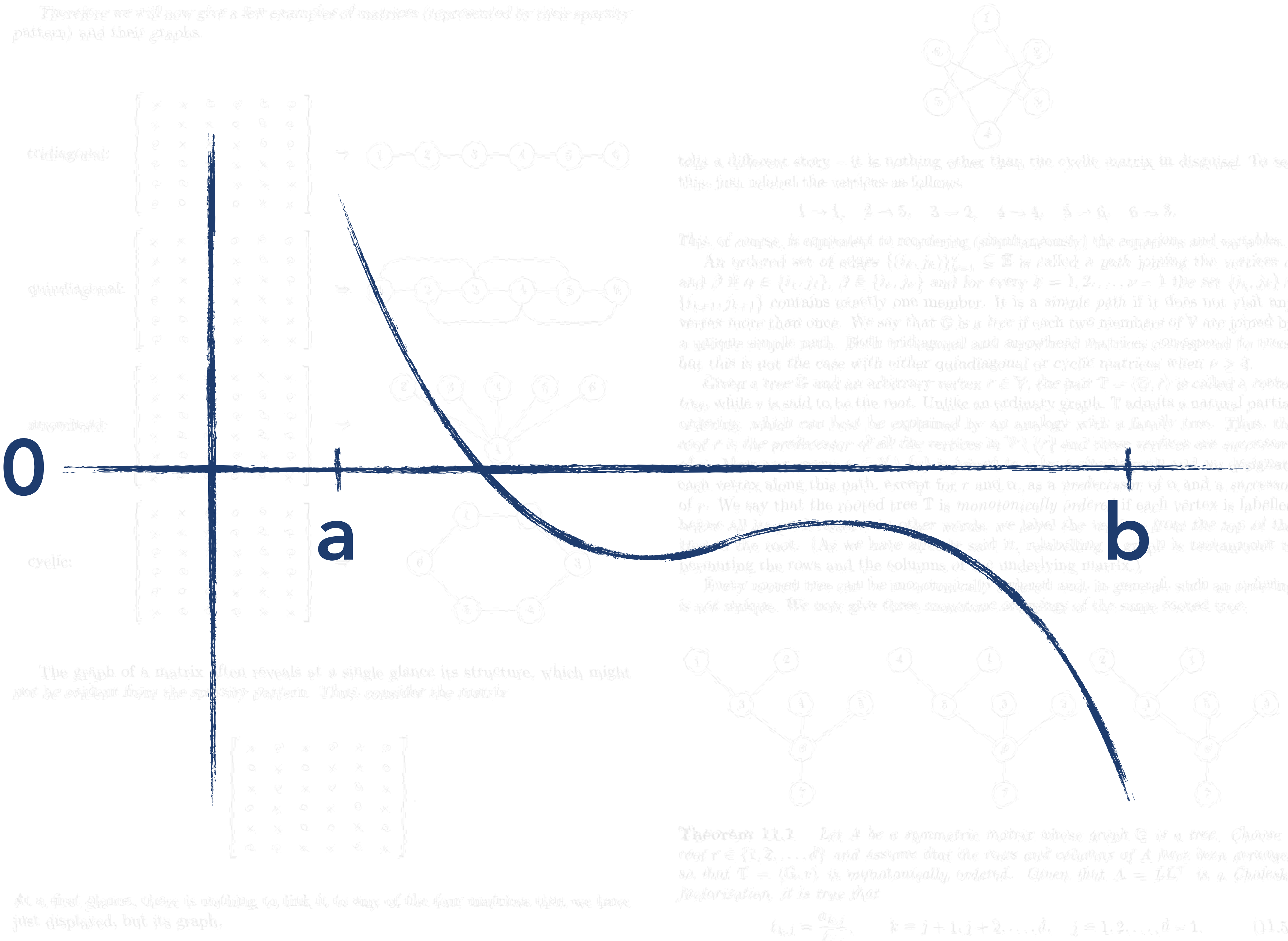
5

1 PICTURE = 1000 WORDS

```
// TODO: Write tests later
test('it renders', async function(assert) {
  await render(hbs`<ComplexComponent />`);
  assert.ok(true);
});
```

```
import { percySnapshot } from 'ember-percy';
...
// TODO: Write tests later
test('complex workflow', async function(assert) {
  await visit('/complex-page');
  await percySnapshot(assert);
});
```

IF f IS CONTINUOUS IN $[a, b]$, AND IF $f(a)f(b) < 0$, THEN f MUST HAVE A ZERO IN (a, b).


0
a
b

YOU CAN FIND

EQUALLY MANY NUMBERS

BETWEEN 0 AND 1

AS YOU CAN

BETWEEN -∞ AND ∞.

Therefore we will now give a few examples of matrices (represented by their sparsity pattern) and their graphs.

tridiagonal: $\begin{bmatrix} \times & \times & \circ & \circ & \circ & \circ \\ \times & \times & \times & \circ & \circ & \circ \\ \circ & \times & \times & \times & \circ & \circ \\ \circ & \circ & \times & \times & \times & \circ \\ \circ & \circ & \circ & \times & \times & \times \\ \circ & \circ & \circ & \circ & \times & \times \end{bmatrix} \Rightarrow$ ①—②—③—④—⑤—⑥

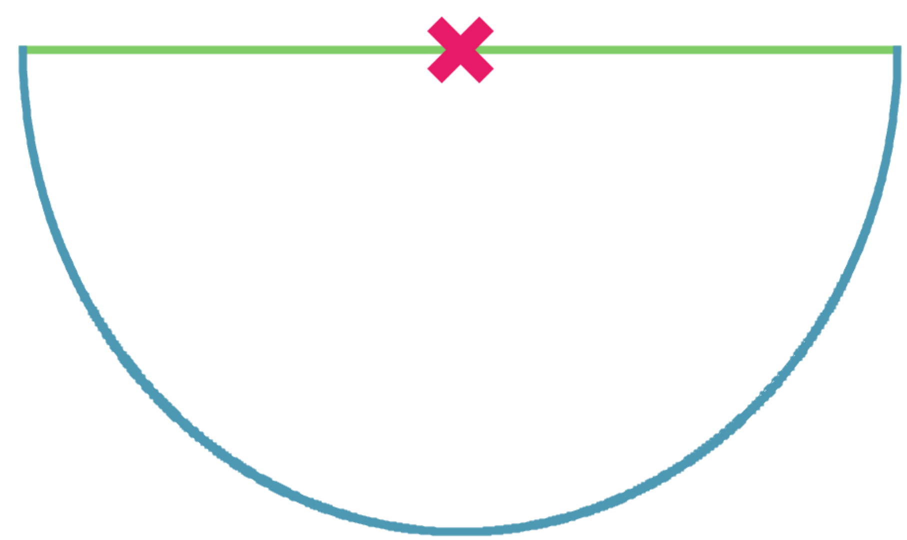
$$\begin{bmatrix} \times & \circ & \times & \circ & \times & \circ \\ \circ & \times & \circ & \times & \times & \circ \\ \times & \circ & \times & \circ & \circ & \times \\ \circ & \times & \circ & \times & \circ & \times \\ \times & \times & \circ & \circ & \times & \circ \\ \circ & \circ & \times & \times & \circ & \times \end{bmatrix}$$

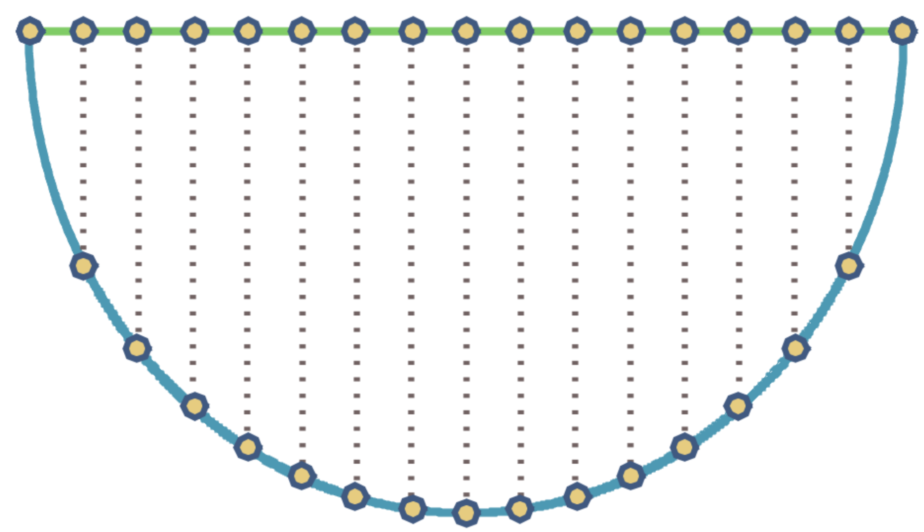
At a first glance, there is nothing to link it to any of the four matrices that we have just displayed, but its graph,

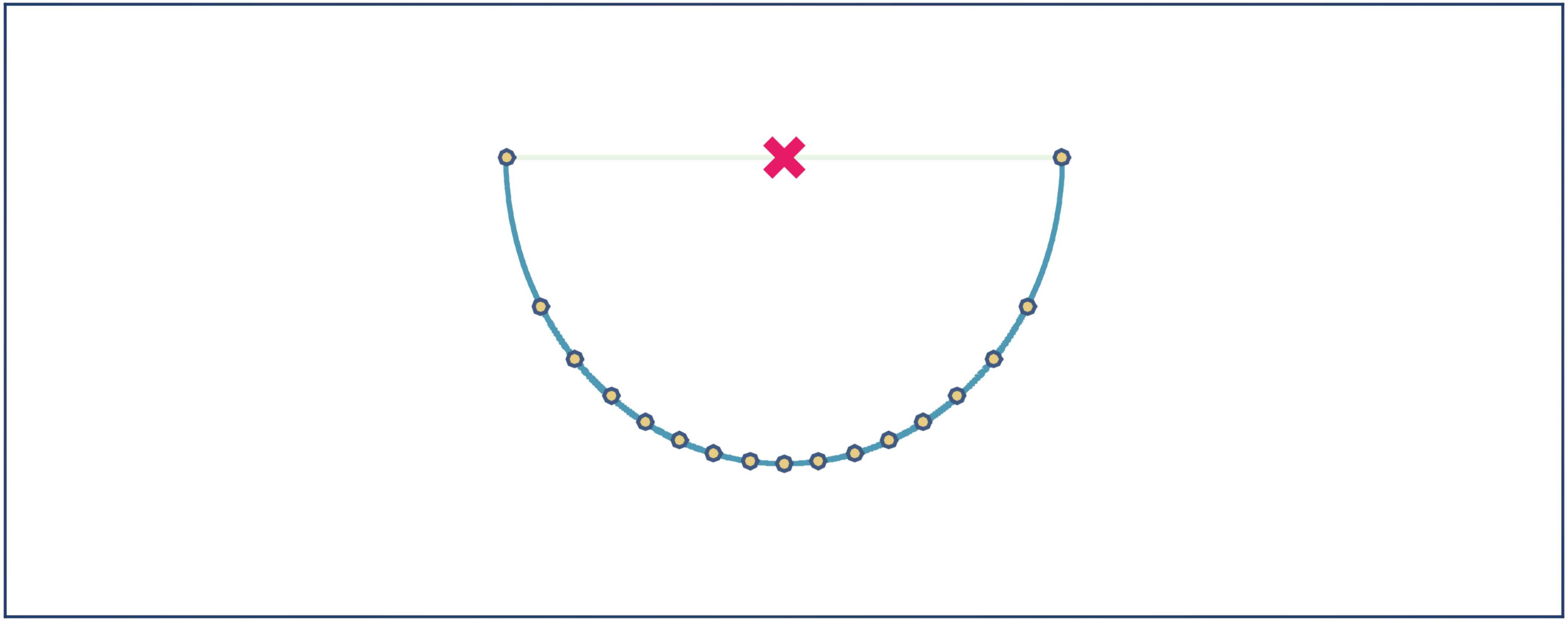
tells a different story – it is nothing other than the cyclic matrix in disguise! To see this, just relabel the vertices as follows:

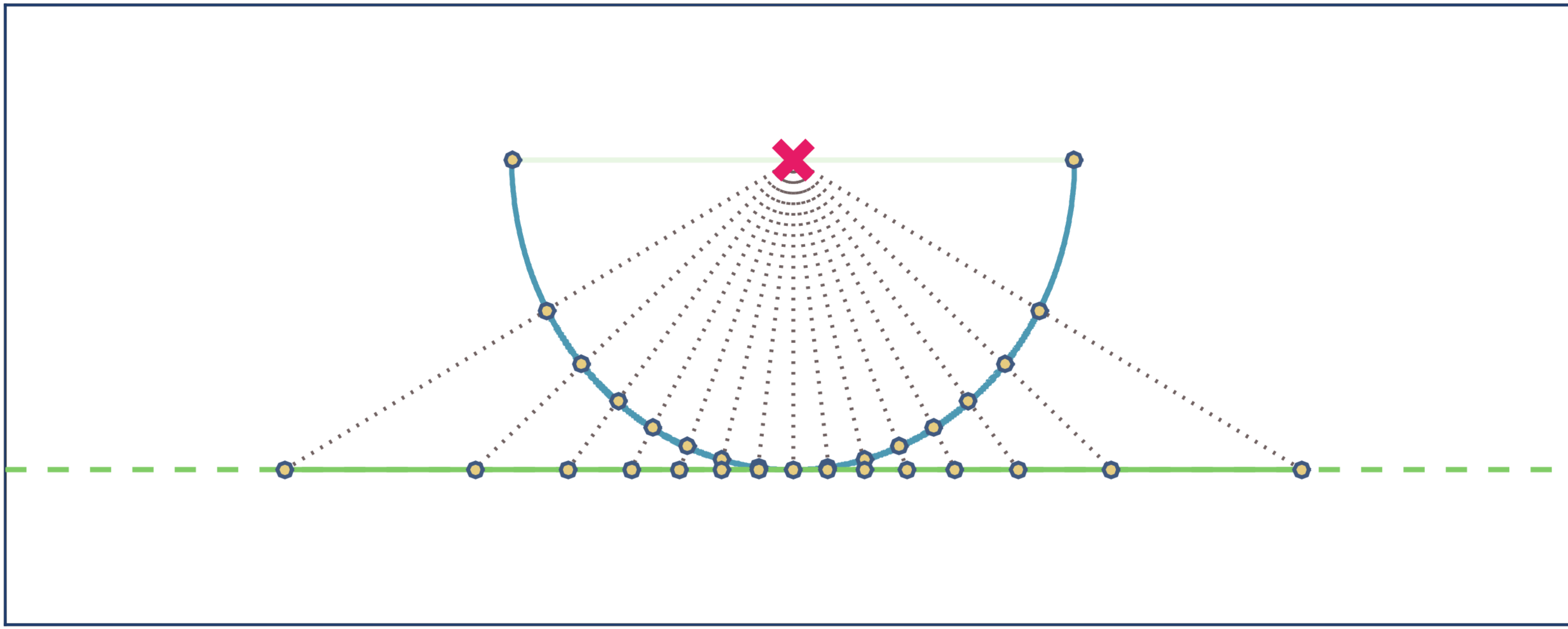
Theorem 11.1 *Let A be a symmetric matrix whose graph $\mathbb{G}$ is a tree. Choose a root $r \in \{1, 2, \ldots, d\}$ and assume that the rows and columns of A have been arranged so that $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is monotonically ordered. Given that $A = LL^\top$ is a Cholesky factorization, it is true that*

$$\ell_{k,j} = \frac{a_{k,j}}{\ell_{j,j}}, \qquad k = j+1, j+2, \ldots, d, \quad j = 1, 2, \ldots, d-1. \tag{11.5}$$









Therefore we will now give a few examples of matrices (represented by their sparsity pattern) and their graphs.

tridiagonal: $\begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} \Rightarrow$

$$\begin{bmatrix} \times & 0 & \times & 0 & \times & 0 \\ 0 & \times & 0 & \times & \times & 0 \\ \times & 0 & \times & 0 & 0 & \times \\ 0 & \times & 0 & \times & 0 & \times \\ \times & \times & 0 & 0 & \times & 0 \\ 0 & 0 & \times & \times & 0 & \times \end{bmatrix}.$$

At a first glance, there is nothing to link it to any of the four matrices that we have just displayed, but its graph

tells a different story – it is nothing other than the cyclic matrix in disguise. To see this, just relabel the vertices as follows

Theorem 11.1 *Let A be a symmetric matrix whose graph $\mathbb{G}$ is a tree. Choose a root $r \in \{1, 2, \ldots, d\}$ and assume that the rows and columns of A have been arranged so that $\mathbb{T} = \langle \mathbb{G}, r \rangle$ is monotonically ordered. Given that $A = LL^T$ is a Cholesky factorization, it is true that*

$$\ell_{k,j} = \frac{a_{k,j}}{\ell_{j,j}}, \qquad k = j+1, j+2, \ldots, d, \quad j = 1, 2, \ldots, d-1. \tag{11.5}$$

5 Rules of Writing Tests

Filter:

Module: Acceptance | Outline ▼

EmberFest 18.10.2019

Running:

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

1 assertion of 1 passed, 0 failed.

If, if, if

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

2 assertions of 2 passed, 0 failed.

- If, if, if
- Use common, everyday words

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

3 assertions of 3 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline

EmberFest 18.10.2019

4 assertions of 4 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms
- All your basis are belong to us

5 Rules of Writing Tests

Filter: Module: Acceptance | Outline ▼

EmberFest 18.10.2019

5 assertions of 5 passed, 0 failed.

- If, if, if
- Use common, everyday words
- Write less with theorems and new terms
- All your basis are belong to us
- 1 picture = 1000 words

crunchingnumbers.live

@ijlee2

Q.E.D.