Structure-Aware Cover-Free Families

Thaís Bardini Idalino
Universidade Federal de Santa Catarina - Brazil

Stinson66 - New Advances in Designs, Codes and Cryptography

uOttawa

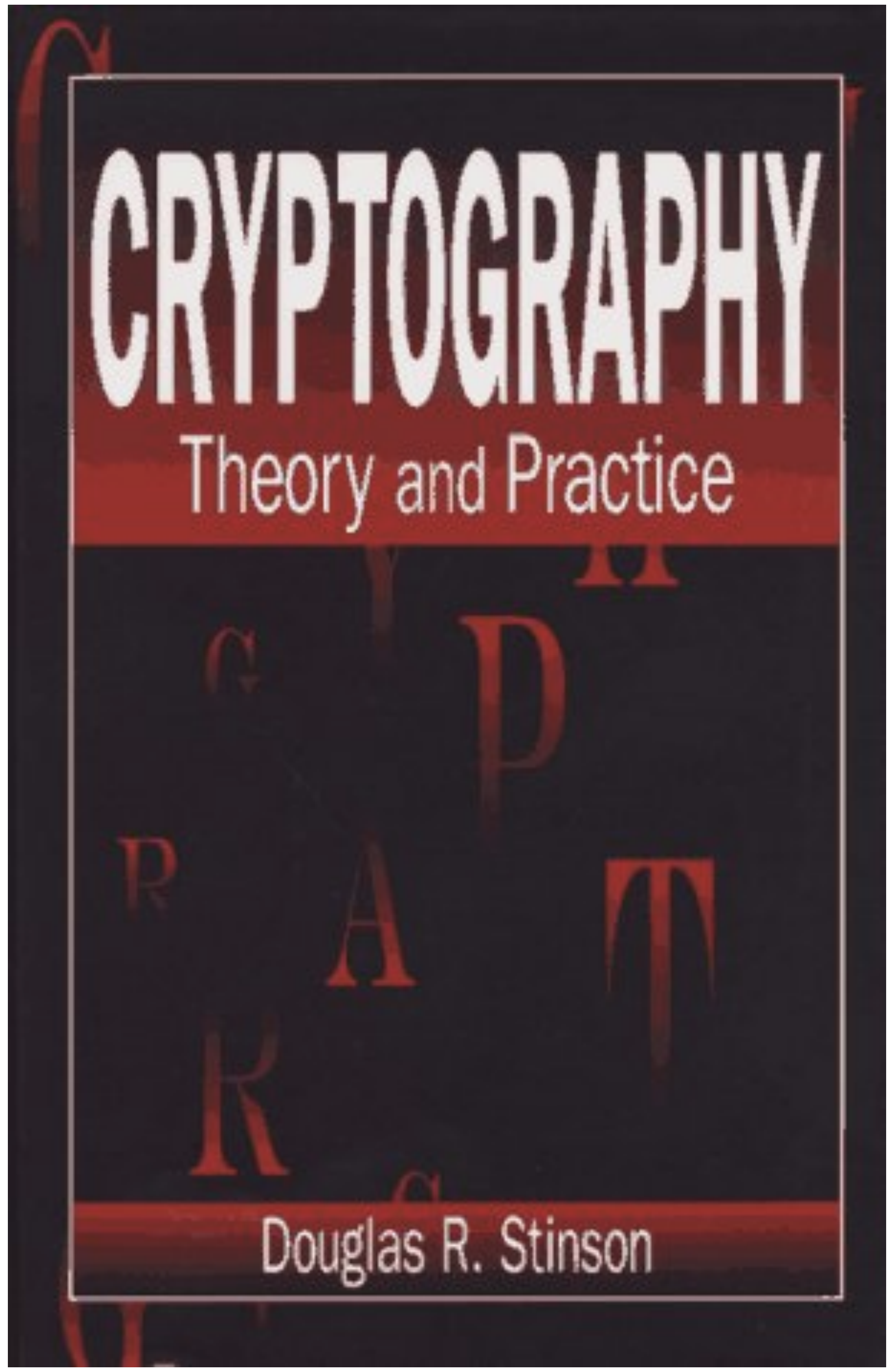
UFSC

Joint work with Lucia Moura

CRYPTOGRAPHY

Theory and Practice

Douglas R. Stinson



Group Testing and Batch Verification

Gregory M. Zaverucha and Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzaveruc,dstinson}@uwaterloo.ca

Abstract. We observe that finding invalid signatures in batches of signatures that fail batch verification is an instance of the classical group testing problem. We survey relevant group testing techniques, and present and compare new sequential and parallel algorithms for finding invalid signatures based on group testing algorithms. Of the five new algorithms, three show improved performance for many parameter choices, and the performance gains are especially notable when multiple processors are available.

Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE @ DIRECT•
Discrete Mathematics 279 (2004) 463–477

DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Generalized cover-free families

D.R. Stinson[a], R. Wei[b]

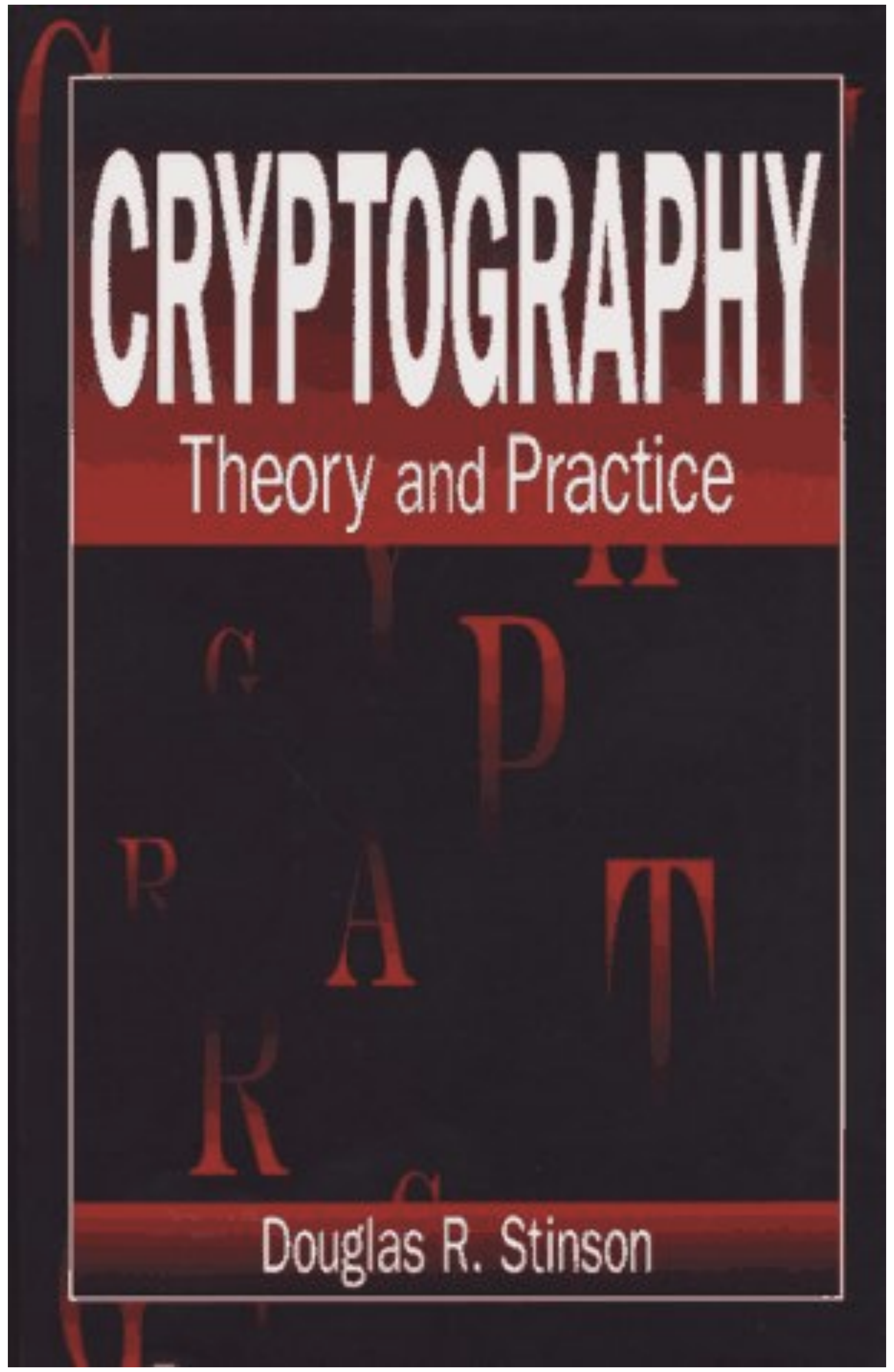
[a] School of Computer Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1
[b] Department of Computer Science, Lakehead University, Thunder Bay, Ont., Canada P7B 5E1

Received 15 November 2002; received in revised form 5 April 2003; accepted 9 June 2003

Abstract

Cover-free families have been investigated by many researchers, and several variations of these set systems have been used in diverse applications. In this paper, we introduce a generalization of cover-free families which includes as special cases all of the previously used definitions. Then we give several bounds and some efficient constructions for these generalized cover-free families.

Keywords: Cover-free family; Probabilistic method



Group Testing and Batch Verification

Gregory M. Zaverucha and Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzaveruc,dstinson}@uwaterloo.ca

Abstract. We observe that finding invalid signatures in batches of signatures that fail batch verification is an instance of the classical group testing problem. We survey relevant group testing techniques, and present and compare new sequential and parallel algorithms for finding invalid signatures based on group testing algorithms. Of the five new algorithms, three show improved performance for many parameter choices, and the performance gains are especially notable when multiple processors are available.

Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE @ DIRECT®
Discrete Mathematics 279 (2004) 463–477

DISCRETE MATHEMATICS
www.elsevier.com/locate/disc

Generalized cover-free families

D.R. Stinson[a], R. Wei[b]

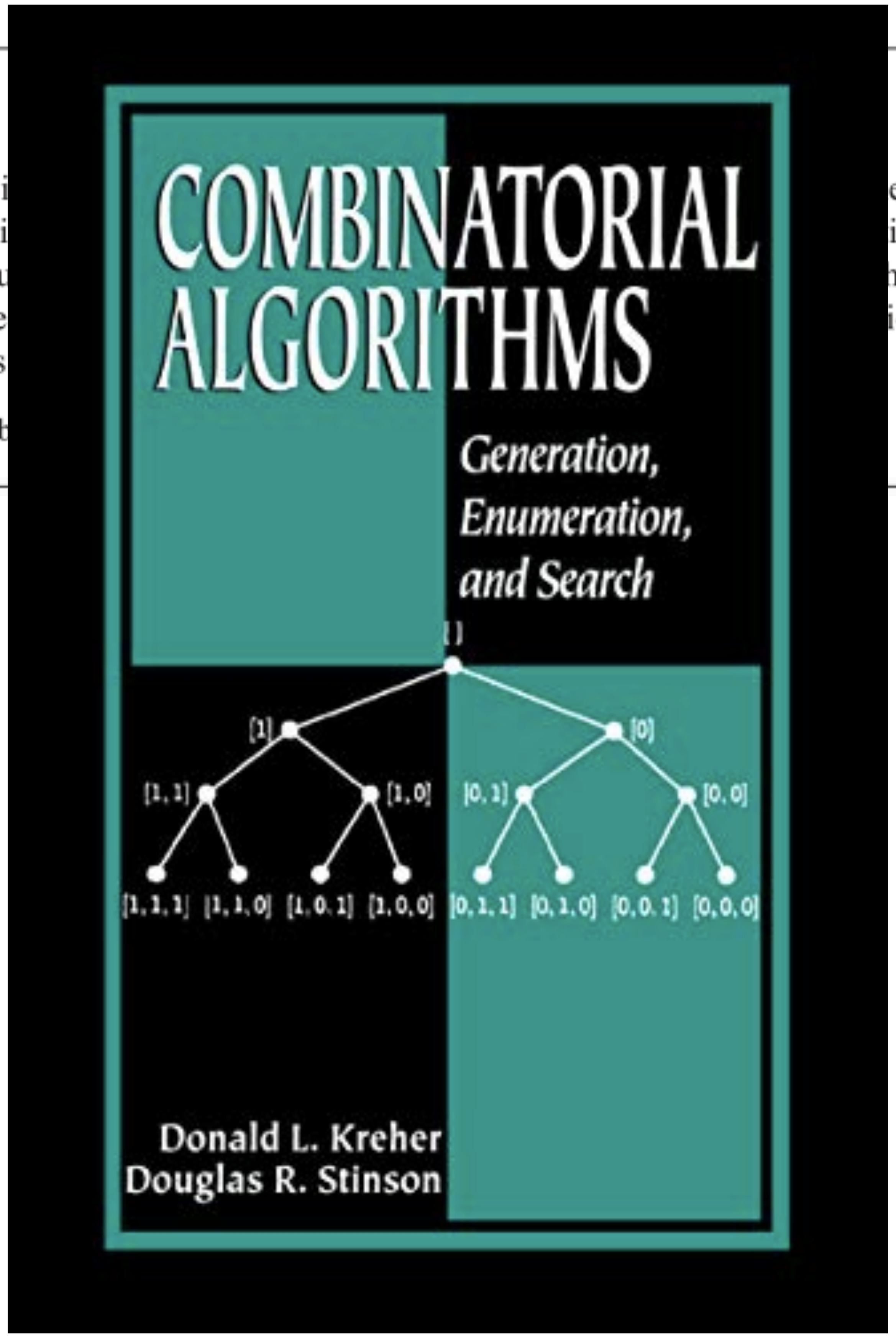
[a] School of Computer Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1
[b] Department of Computer Science, Lakehead University, Thunder Bay, Ont., Canada P7B 5E1

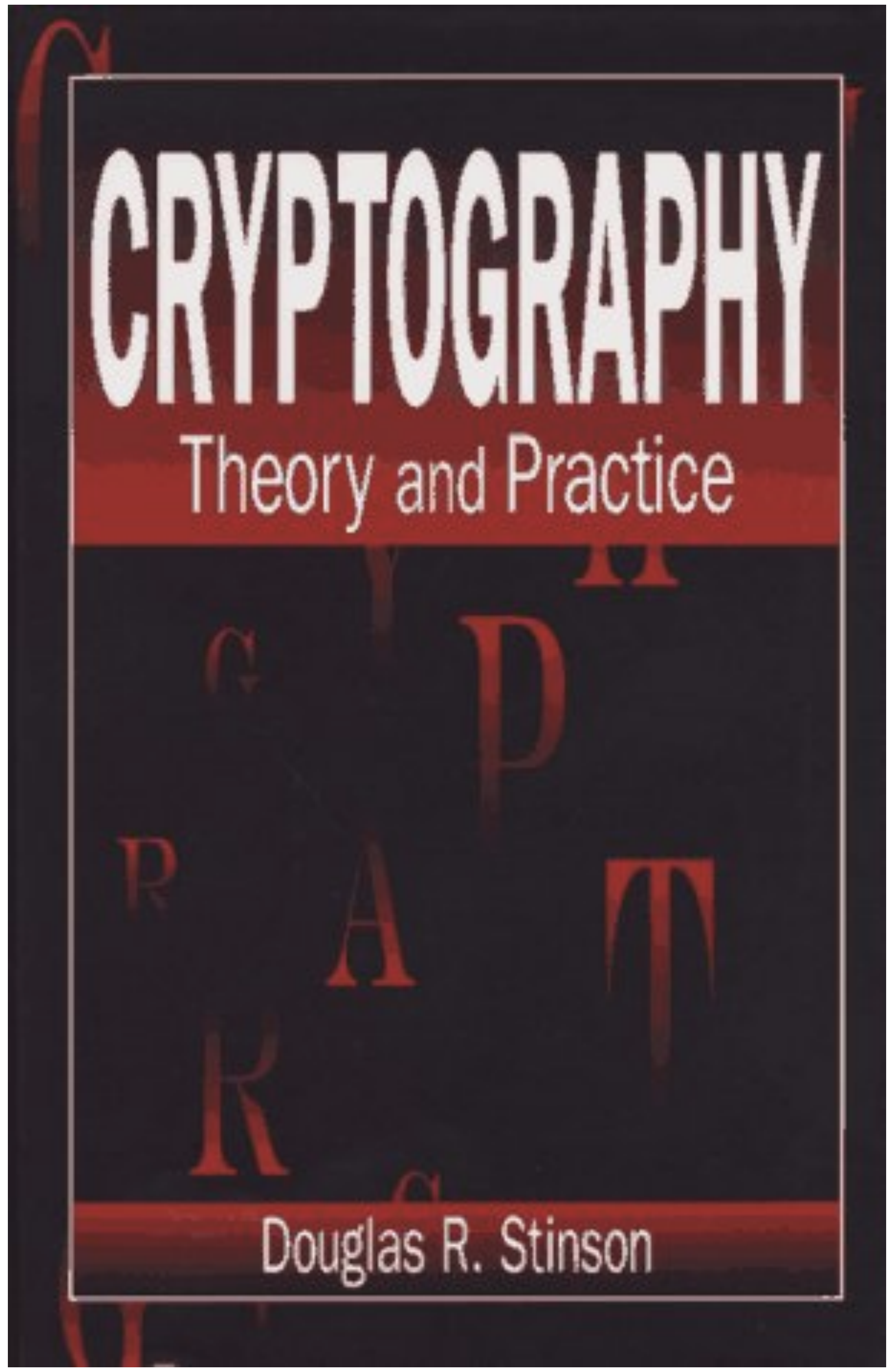
Received 15 November 2002; received in revised form 5 April 2003; accepted 9 June 2003

Abstract

Cover-free families have been i … ese
set systems have been used in di … ion
of cover-free families which inclu … hen
we give several bounds and some … ies.

Keywords: Cover-free family; Probab





Group Testing and Batch Verification

Gregory M. Zaverucha and Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{gzaveruc,dstinson}@uwaterloo.ca

Abstract. We observe that finding invalid signatures in batches of signatures that fail batch verification is an instance of the classical group testing problem. We survey relevant group testing techniques, and present and compare new sequential and parallel algorithms for finding invalid signatures based on group testing algorithms. Of the five new algorithms, three show improved performance for many parameter choices, and the performance gains are especially notable when multiple processors are available.

ELSEVIER

Available at
www.ElsevierMathematics.com
POWERED BY SC

DISCRETE

Discrete Mathematics 2

Generalized c

D.R. Stins
[a]School of Computer Science, University
[b]Department of Computer Science, Lakehead

Received 15 November 2002; received in

Abstract

Cover-free families have been i
set systems have been used in di
of cover-free families which inclu
we give several bounds and some

Keywords: Cover-free family; Probab

atch Verification

Douglas R. Stinson

f Computer Science
Vaterloo
3G1, Canada
@uwaterloo.ca

alid signatures in batches of sig-
n instance of the classical group
p testing techniques, and present
el algorithms for finding invalid
hms. Of the five new algorithms,
nany parameter choices, and the
le when multiple processors are
available.

Structure-Aware Cover-Free Families

Thaís Bardini Idalino
Universidade Federal de Santa Catarina - Brazil

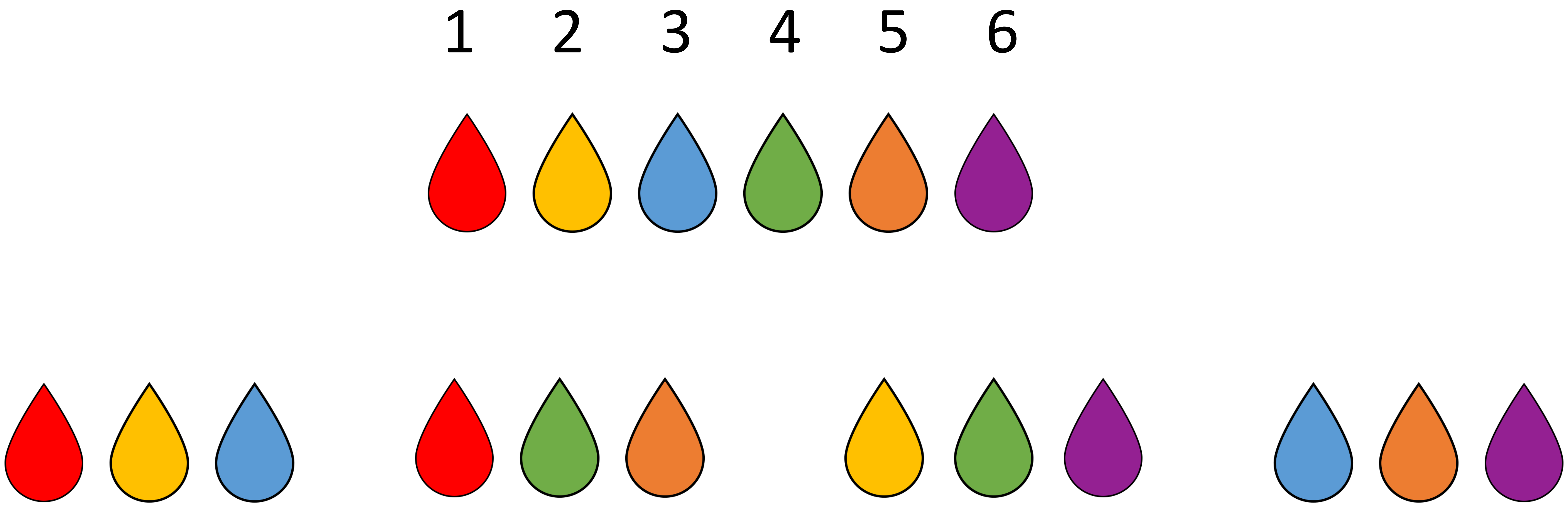
uOttawa

UFSC

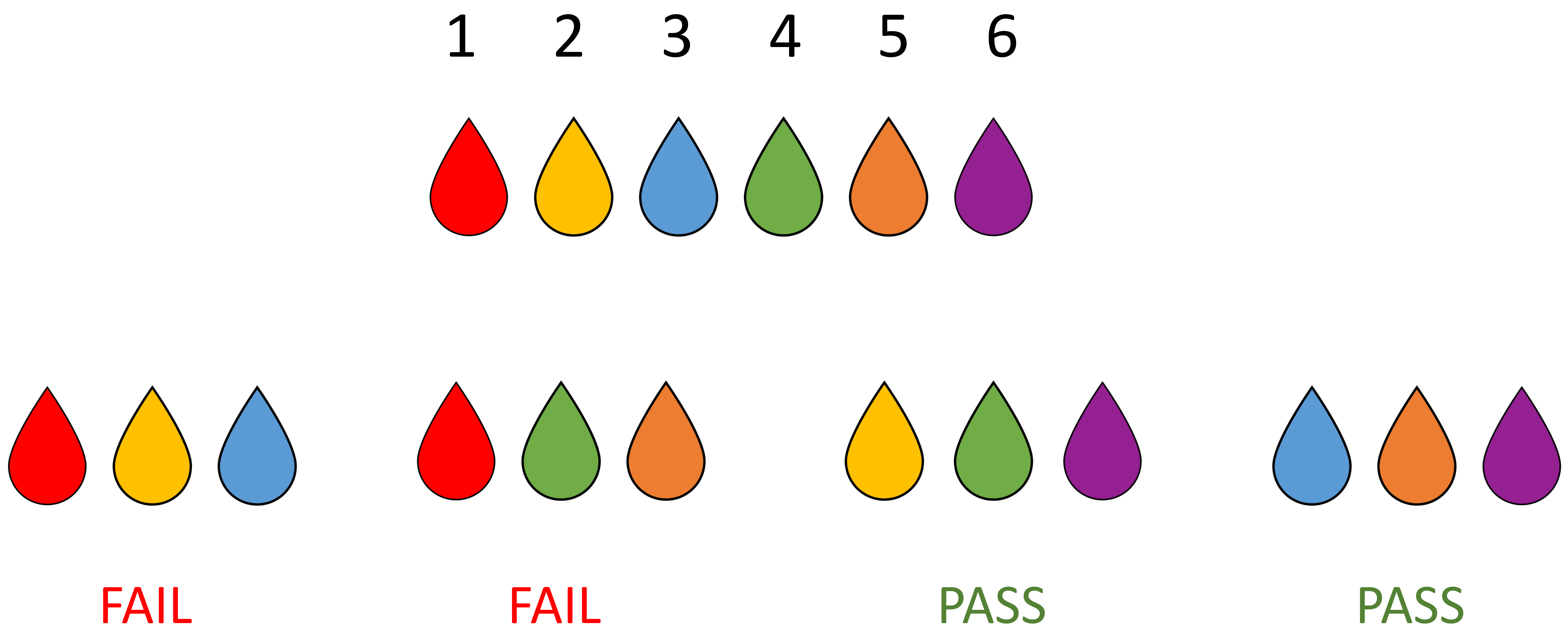
Stinson66 - New Advances in Designs, Codes and Cryptography

Joint work with Lucia Moura

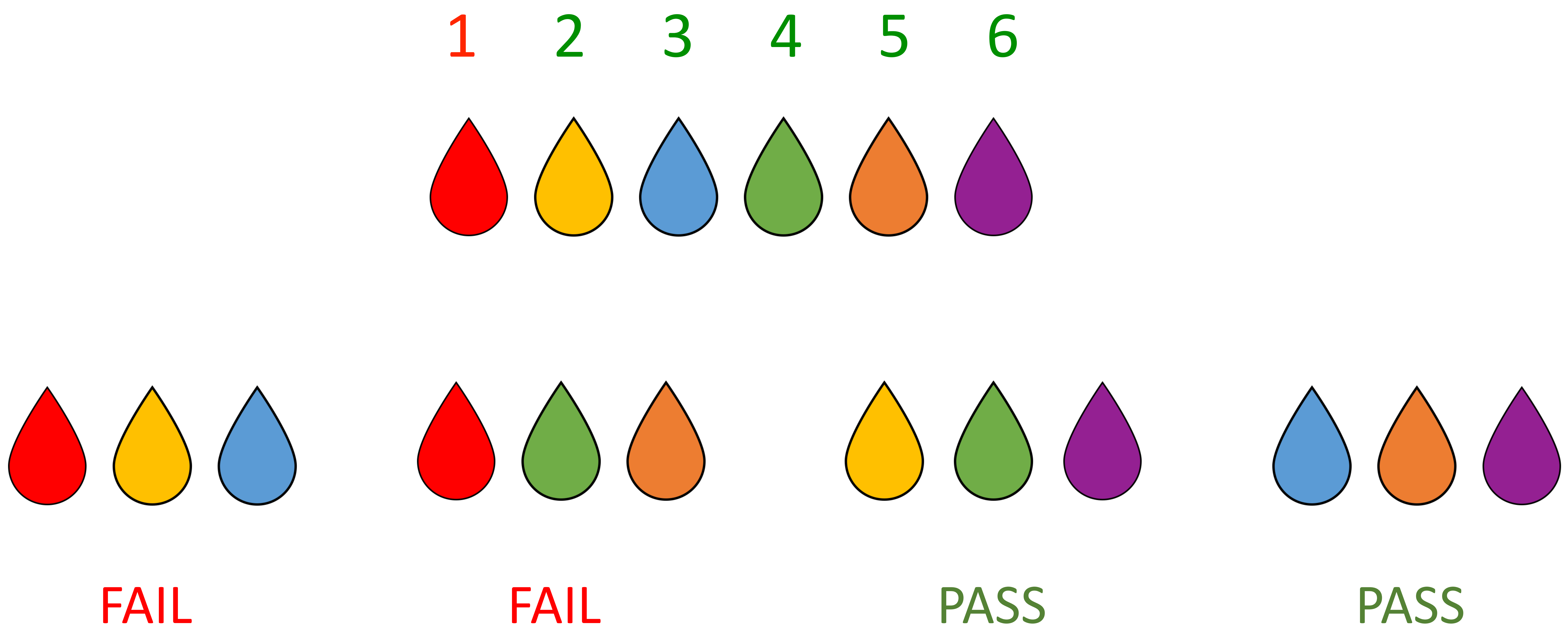
Combinatorial Group Testing



Combinatorial Group Testing

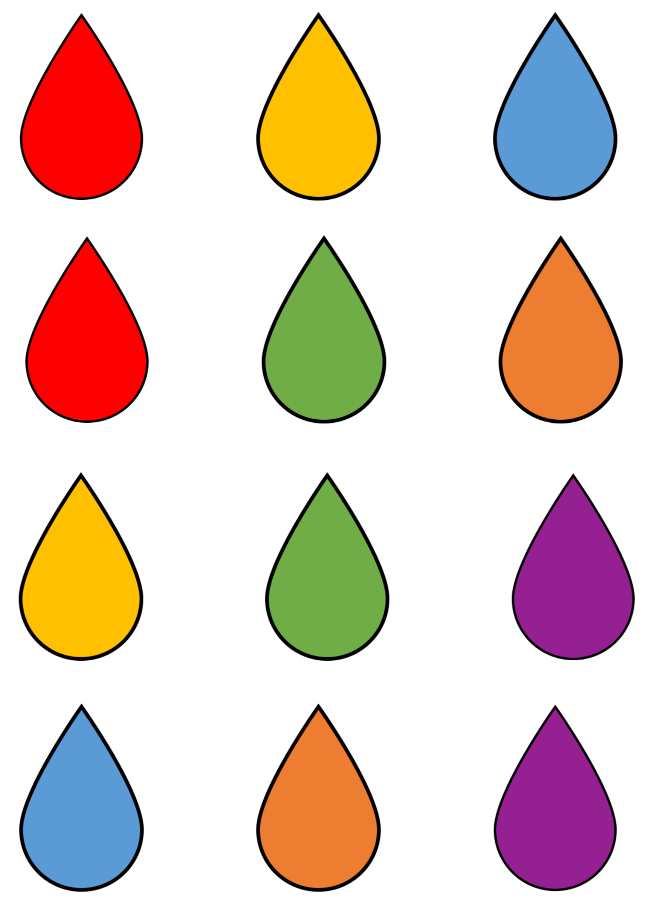


Combinatorial Group Testing



Cover-Free Families

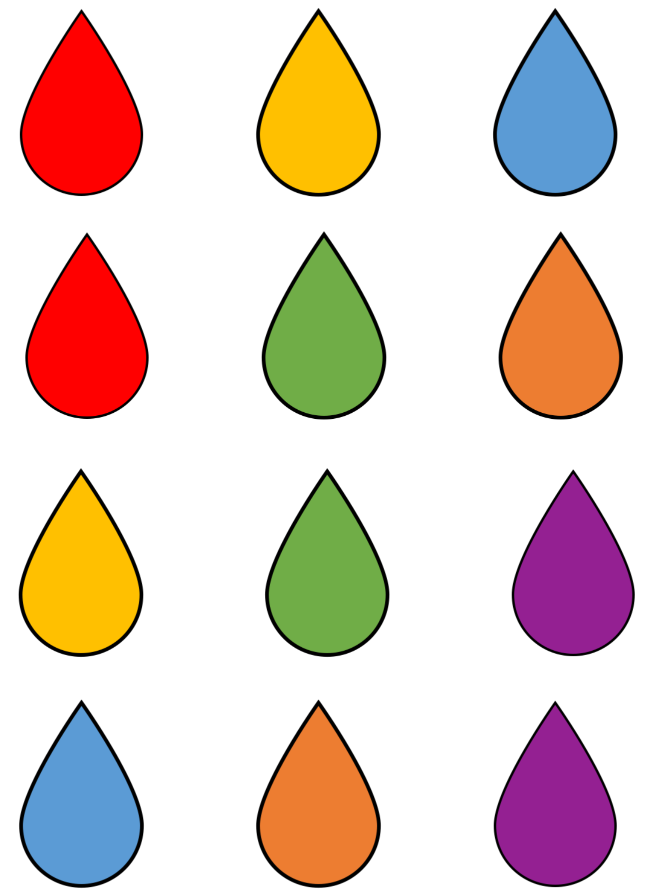
	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL
FAIL
PASS
PASS

Cover-Free Families

	1	2	3	4	5	6	
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS



d – CFF(t, n)

Cover-Free Families

Definition: Let d be a positive integer. A *d*-cover-free family, denoted $d - CFF(t, n)$, is a set system $\mathcal{F} = (X, \mathcal{B})$ with $|X| = t$ and $|\mathcal{B}| = n$ such that for any $d + 1$ subsets $B_{i_0}, B_{i_1}, \ldots, B_{i_d} \in \mathcal{B}$, we have:

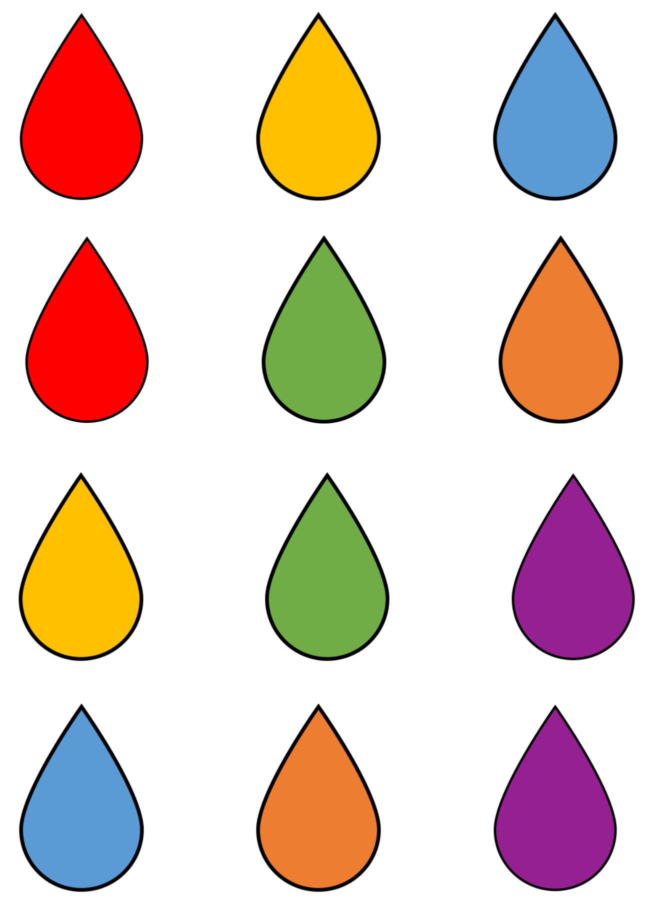
$$\left| B_{i_0} \setminus \left(\bigcup_{j=1}^{d} B_{i_j} \right) \right| \geq 1.$$

No element is ***covered*** by the union of any other d.

* Equivalent to disjunct matrices and superimposed codes.

Cover-Free Families

	1	2	3	4	5	6	
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS



***1* – CFF(4, 6)**

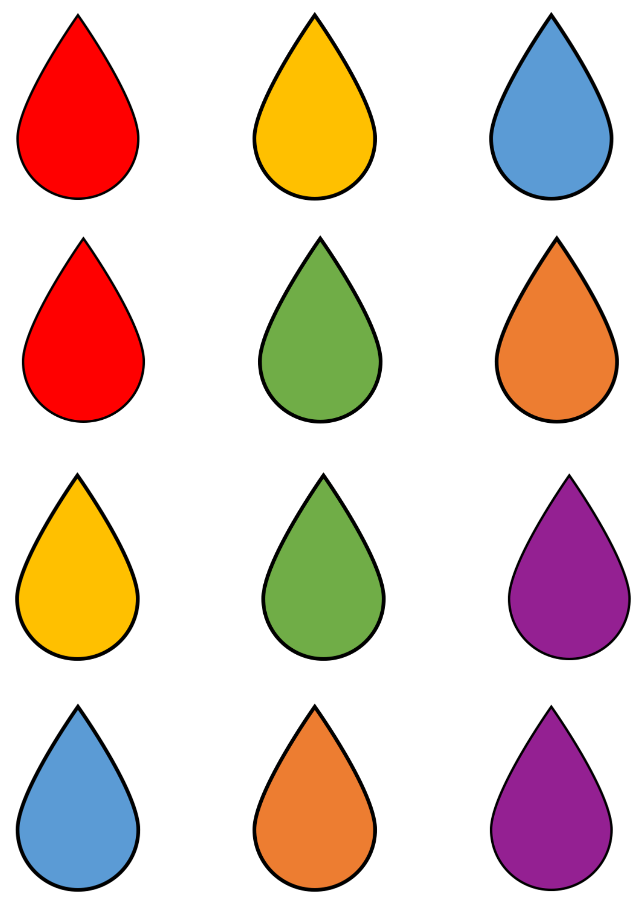
Cover-Free Families

	1	2	3	4	5	6	
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS

***1* – CFF(4, 6)**

Cover-Free Families

	1	2	3	4	5	6
Test 1	1	1	1	0	0	0
Test 2	1	0	0	1	1	0
Test 3	0	1	0	1	0	1
Test 4	0	0	1	0	1	1



FAIL

FAIL

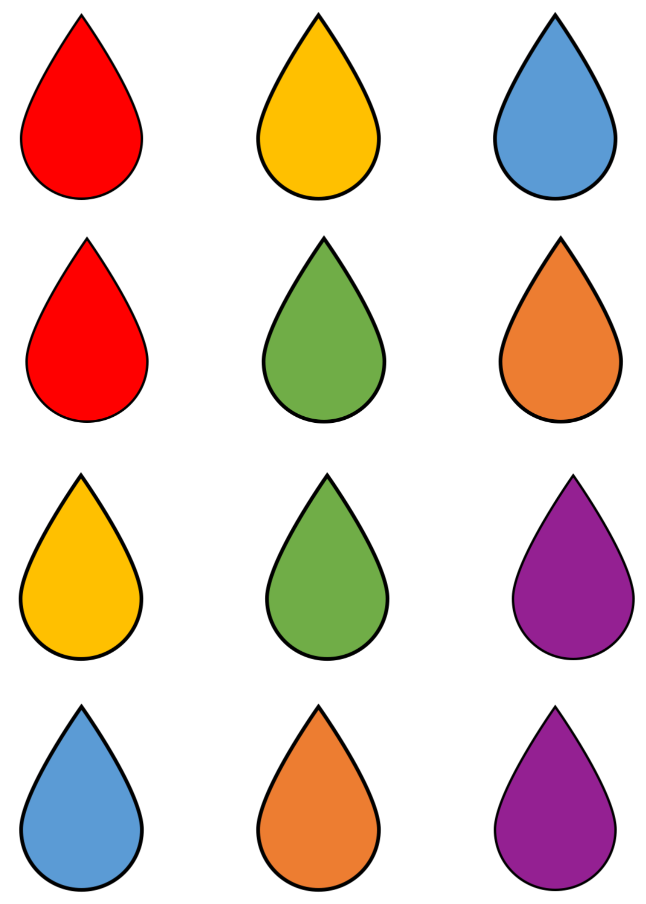
PASS

PASS

***1* – CFF(4, 6)**

Cover-Free Families

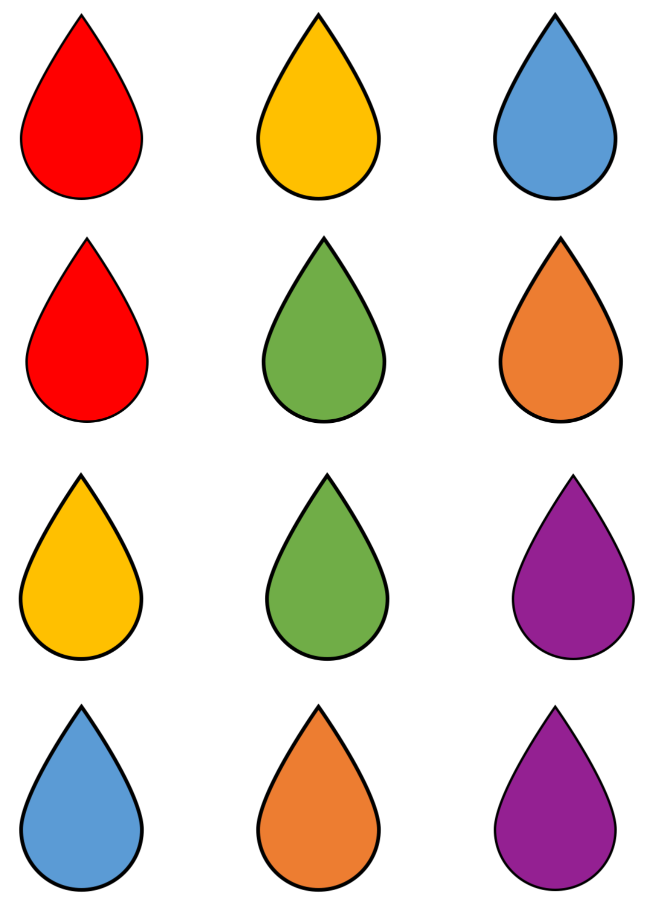
	1	2	3	4	5	6	
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS



***1* – CFF(4, 6)**

Cover-Free Families

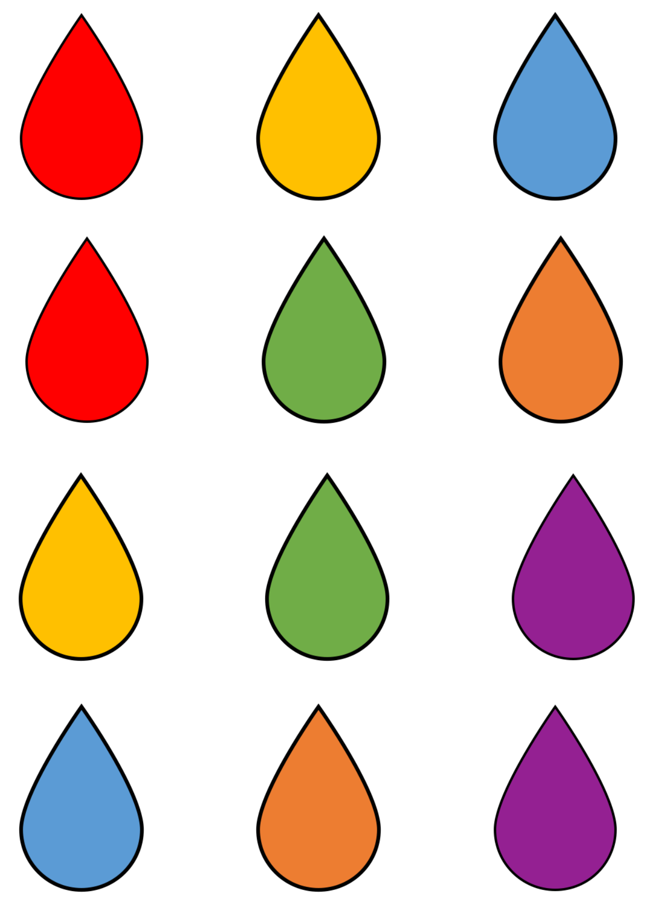
	1	2	3	4	5	6	Result
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS



***1* – CFF(4, 6)**

Cover-Free Families

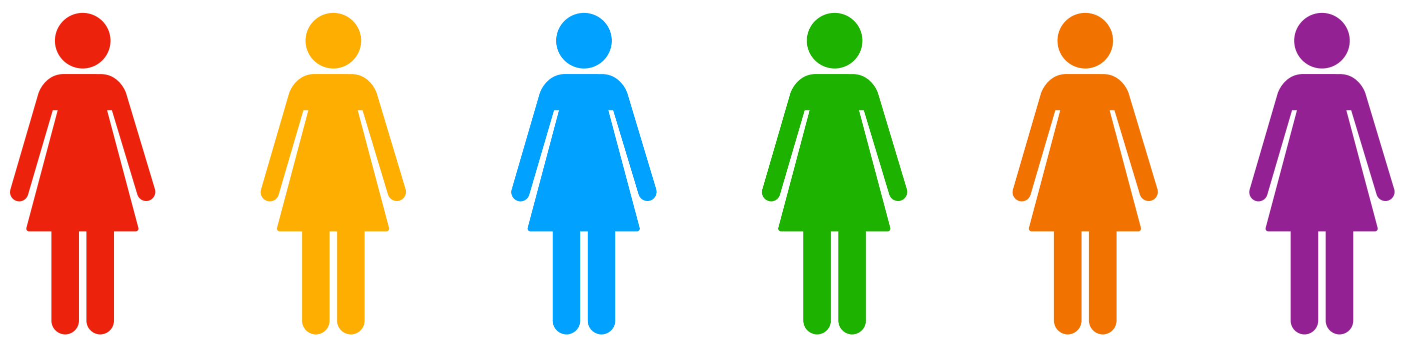
	1	2	3	4	5	6	Result
Test 1	1	1	1	0	0	0	FAIL
Test 2	1	0	0	1	1	0	FAIL
Test 3	0	1	0	1	0	1	PASS
Test 4	0	0	1	0	1	1	PASS



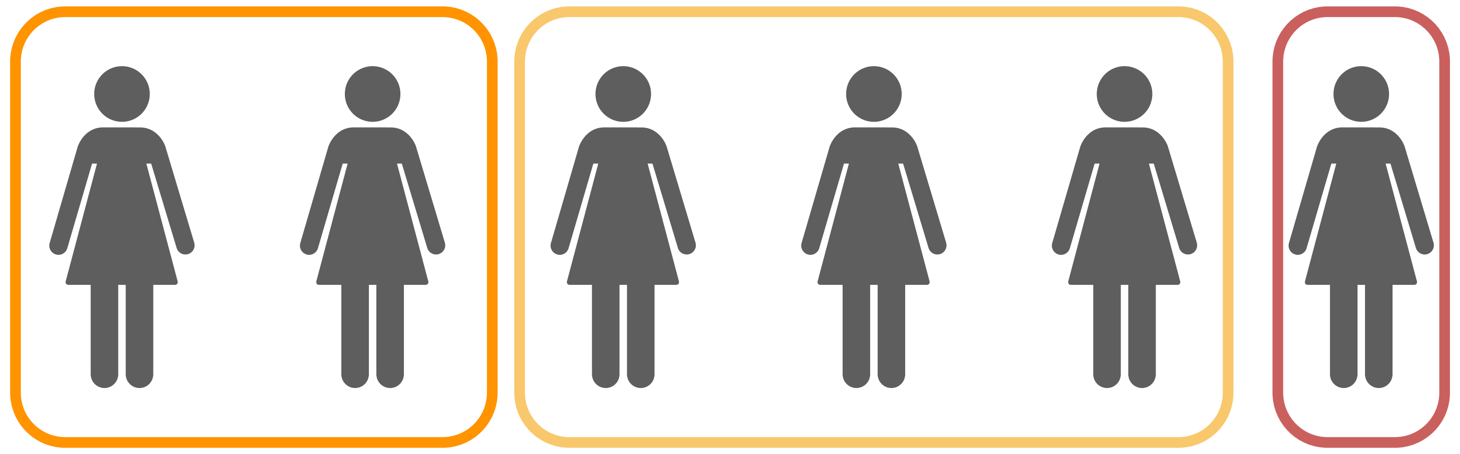
***1* – CFF(4, 6)**

In this talk

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1



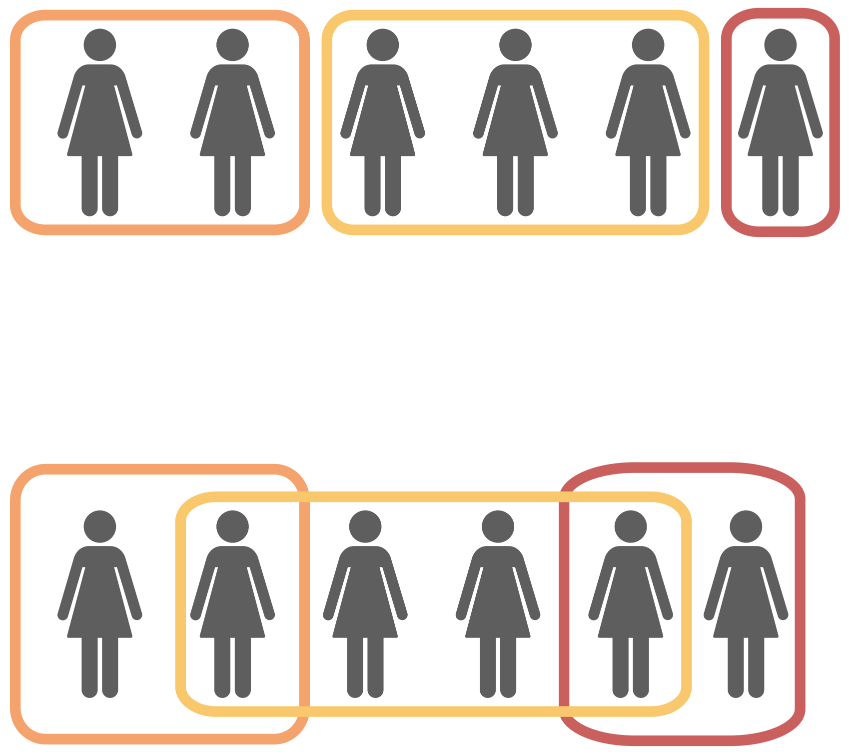
- Applications of **combinatorial group testing** in pandemic screening
- Study of ***structure-aware combinatorial group testing***
- New constructions of structure-aware CFFs
- Examples of applications
- Future work



Structure-aware CFFs

Model the communities as **hypergraphs**

- $\mathscr{H} = (V, \mathscr{S})$

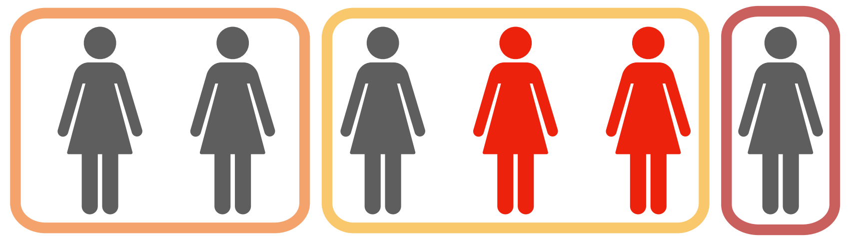


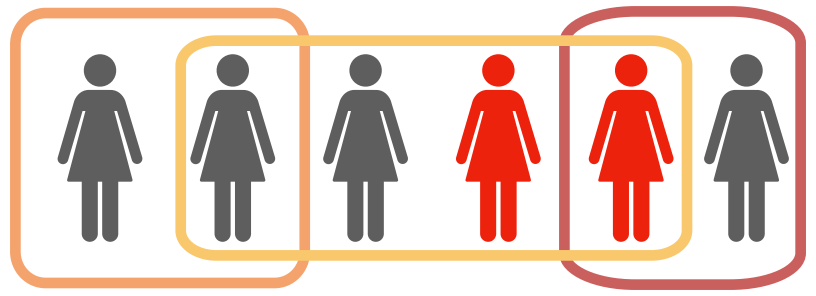
Propose constructions that take $\mathscr{H}$ into consideration

- $(\mathscr{S}, r) - CFF(t, n)$

Structure-aware CFFs

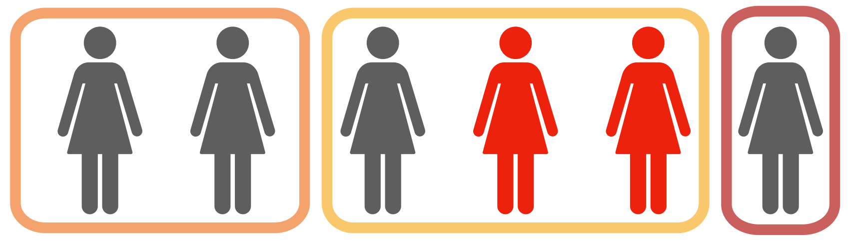
Overlapping and non-overlapping edges:





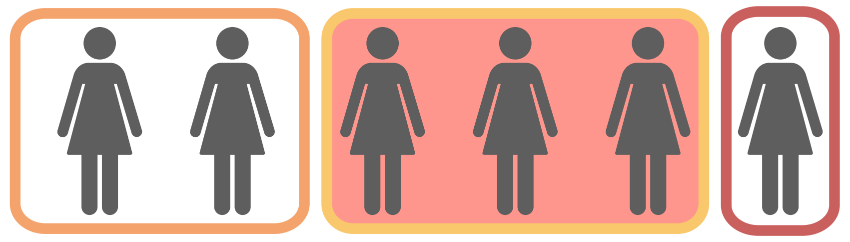
Configurations:

- $(\mathscr{S}, r) - CFF(t, n)$



  - Identify all infected individuals, as long as there are at most *r* infected edges that jointly contain them

- $(\mathscr{S}, r) - ECFF(t, n)$



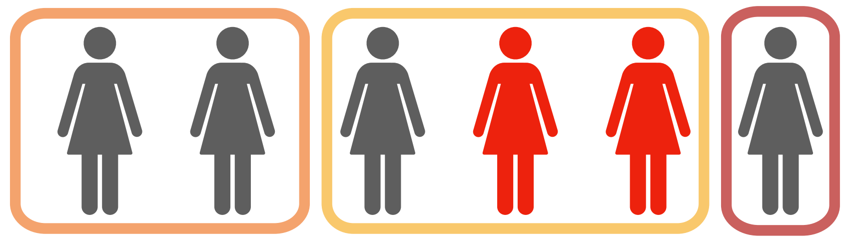
  - Identify *r* infected edges, without internal identification

Related Work

- Several works on CGT for COVID-19 testing
- Few structure-aware solutions (equivalent models to ours)
  - Connected and overlapping communities (Nikolopoulos et al., 2021)
    - Adaptive and non-adaptive algorithms
  - Generalized group testing (Gonen et al., 2022)
    - Edges are all potentially contaminated sets
  - *Variable* CFFs in Cryptography (Idalino, 2019)

This work: Idalino and Moura. Structure-Aware Combinatorial Group Testing: A New Method for Pandemic Screening. IWOCA 2022

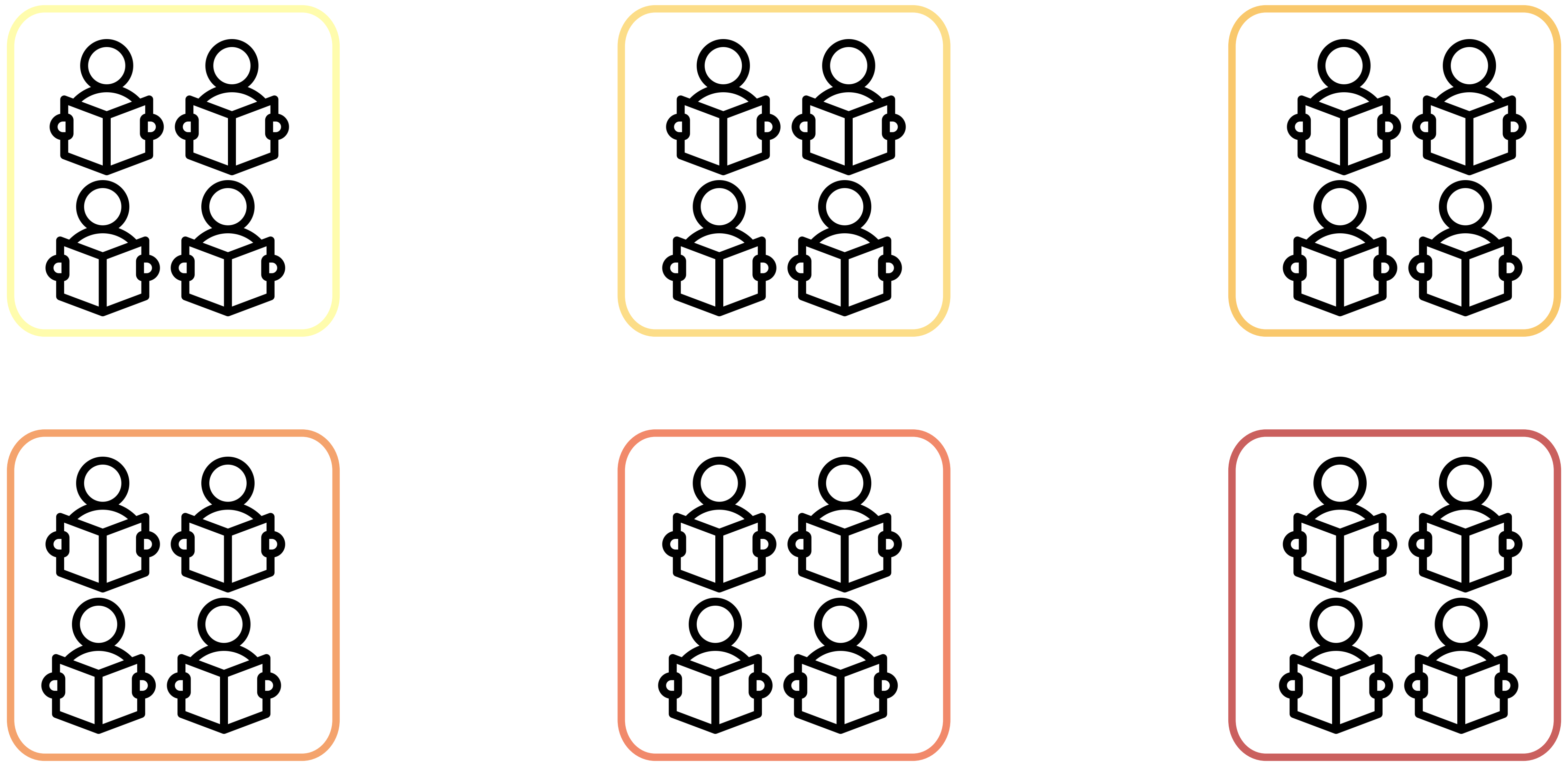
Non-overlapping edges



- Revisit old $d - CFF$ constructions
- Show we can boost the number of infected items they can identify

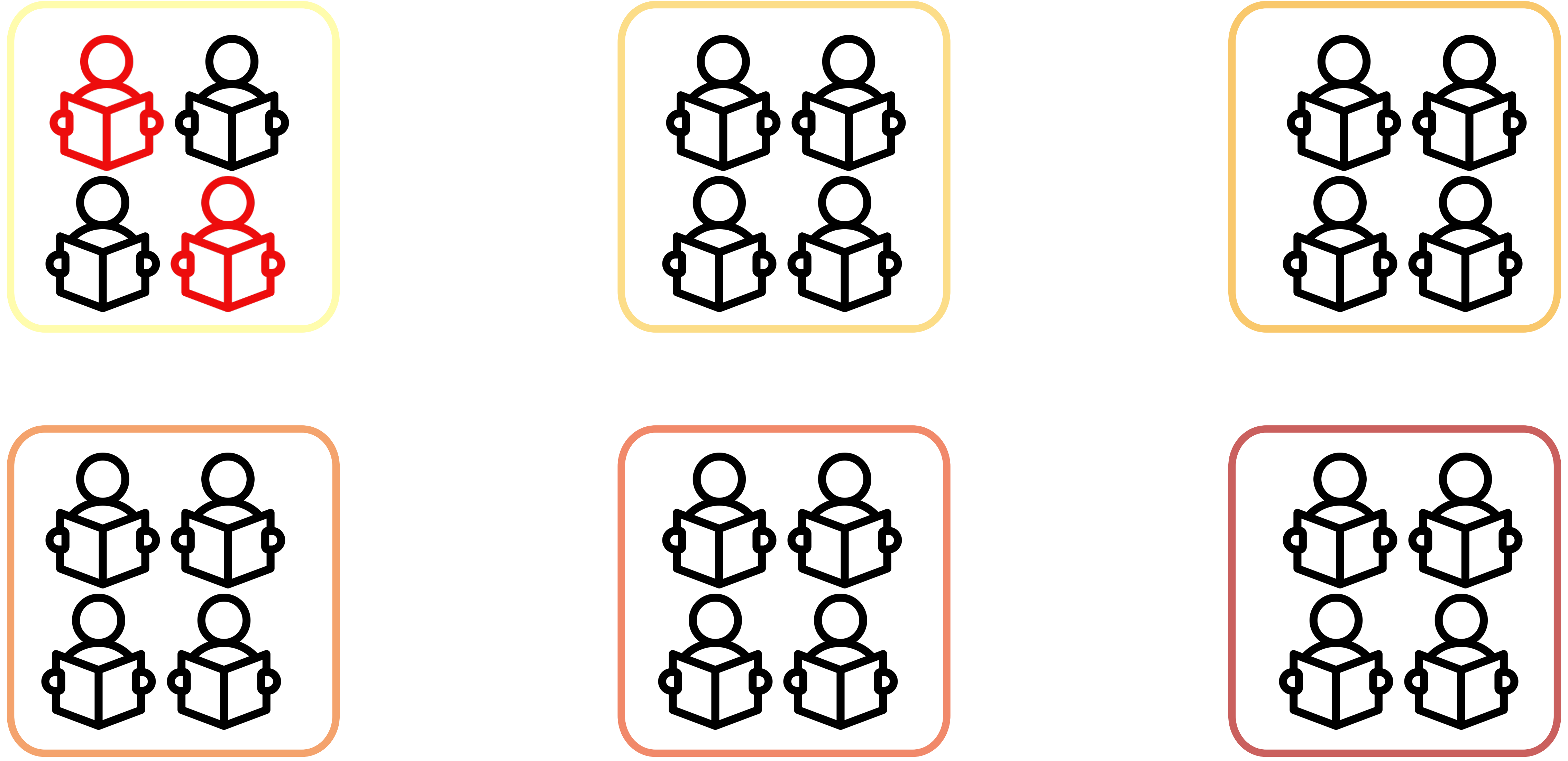
The classroom problem

Non-overlapping edges



The classroom problem

Non-overlapping edges



Sperner-type construction

The classroom problem

One column per classroom

1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

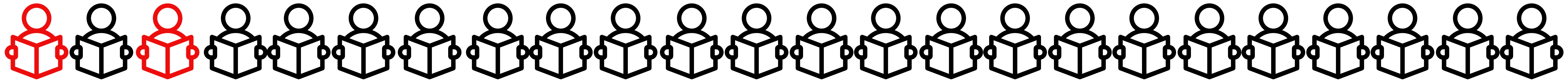
Sperner-type construction

The classroom problem

Classroom 1				Classroom 2				Classroom 3				Classroom 4				Classroom 5				Classroom 6			
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

Sperner-type construction

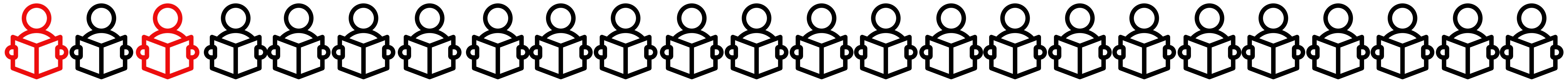
The classroom problem



1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

Sperner-type construction

The classroom problem

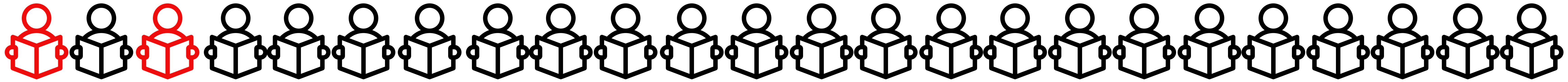


1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	FAIL
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	FAIL
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	PASS
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	PASS

$$(\mathcal{S},1) - ECFF(4,24)$$

Sperner-type construction

The classroom problem

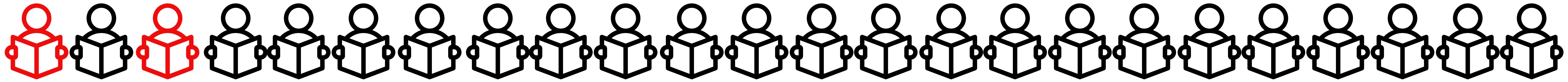


1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1

1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1

Sperner-type construction

The classroom problem



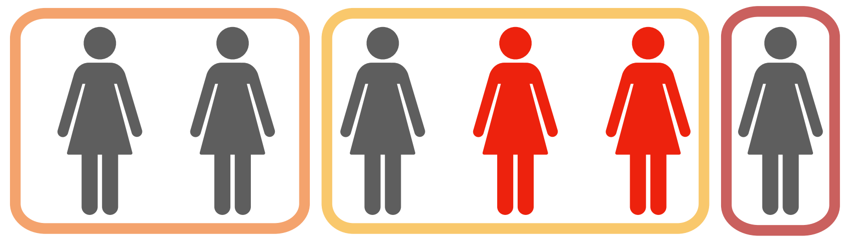
1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	FAIL
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	FAIL
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	PASS
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	PASS

1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	FAIL
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	PASS
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	FAIL
0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	PASS

$$(\mathcal{S},1) - CFF(8,24)$$

Sperner-type construction

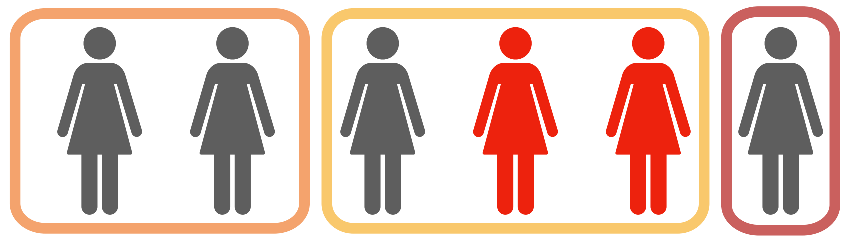
The classroom problem



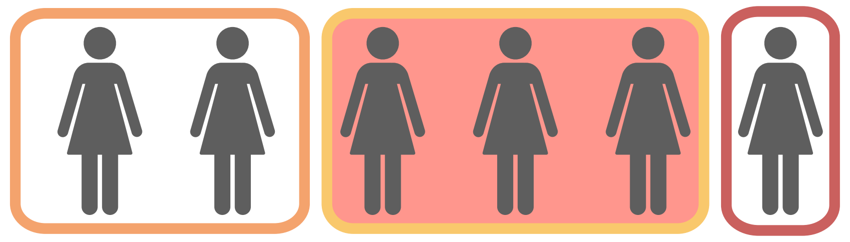
- Consider n individuals divided into m non-overlapping edges, each of size up to d.
- Variation of a $1 - CFF(t_1, m)$ concatenated with a $d \times d$ id-matrix.
  - Generates a $(\mathcal{S}, 1) - CFF(t, n)$, $t = t_1 + d \approx \log m + d = \log n/d + d$

Sperner-type construction

The classroom problem



- Consider n individuals divided into m non-overlapping edges, each of size up to d.
- Variation of a $1 - CFF(t_1, m)$ concatenated with a $d \times d$ id-matrix.
  - Generates a $(\mathcal{S},1) - CFF(t, n)$, $t = t_1 + d \approx \log m + d = \log n/d + d$
- If we only care about infected edges
  - Restrict to the first t_1 rows to get a $(\mathcal{S},1) - ECFF(t_1, n)$



Sperner-type construction

Comparison with traditional $d - CFF(t, n)$

Total number of students

Number of classrooms

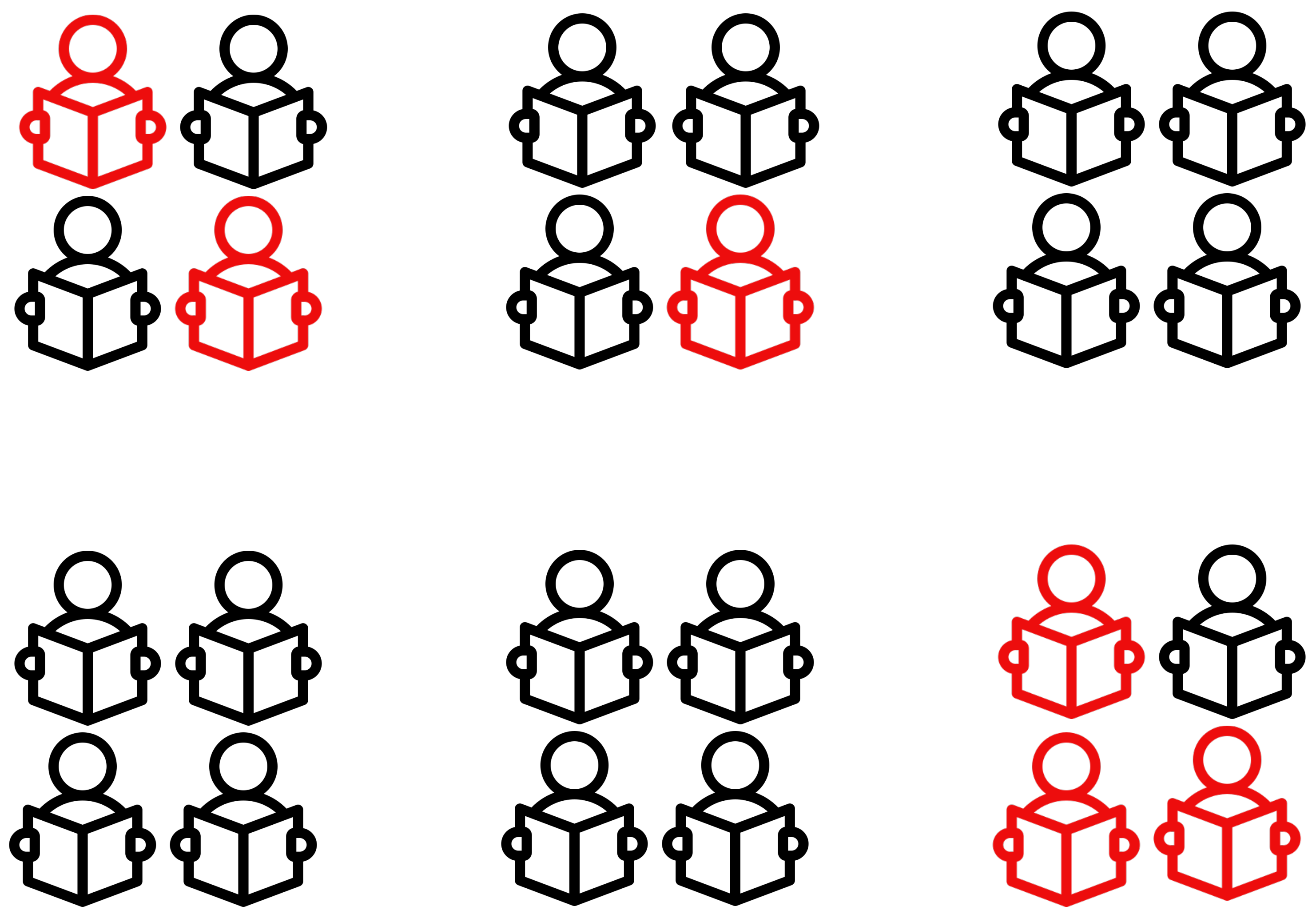
Classroom size

n	m	d	$(\mathcal{S},1) - CFF(t,n)$	$d - CFF(t,n)$
100	10	10	15	66
200	10	20	25	180
300	**10**	**30**	**35**	**231**
100	20	5	11	21
200	20	10	16	66
400	20	20	26	231

Lower bound

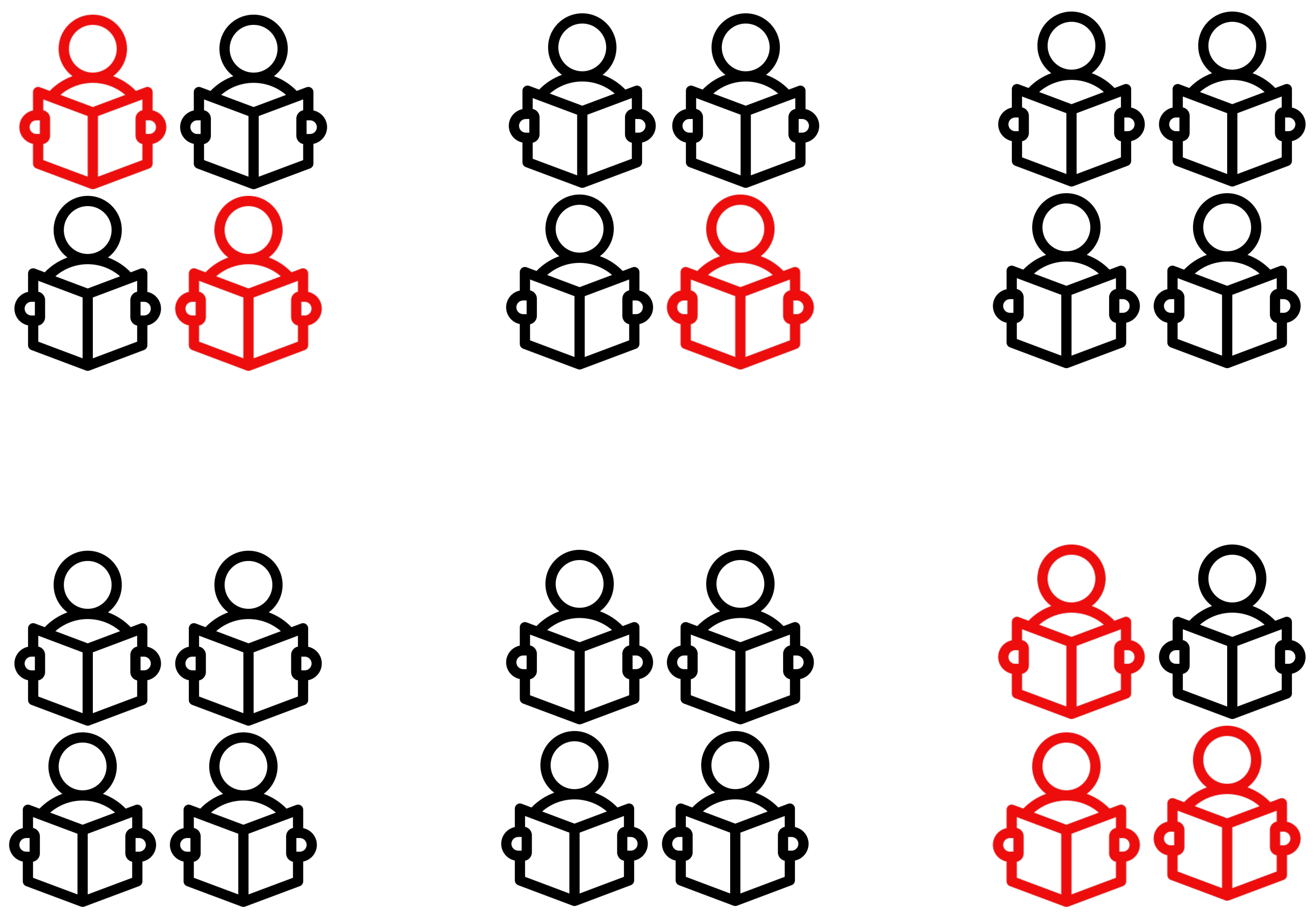
Kronecker-type construction

What if more classrooms are infected?



Kronecker-type construction

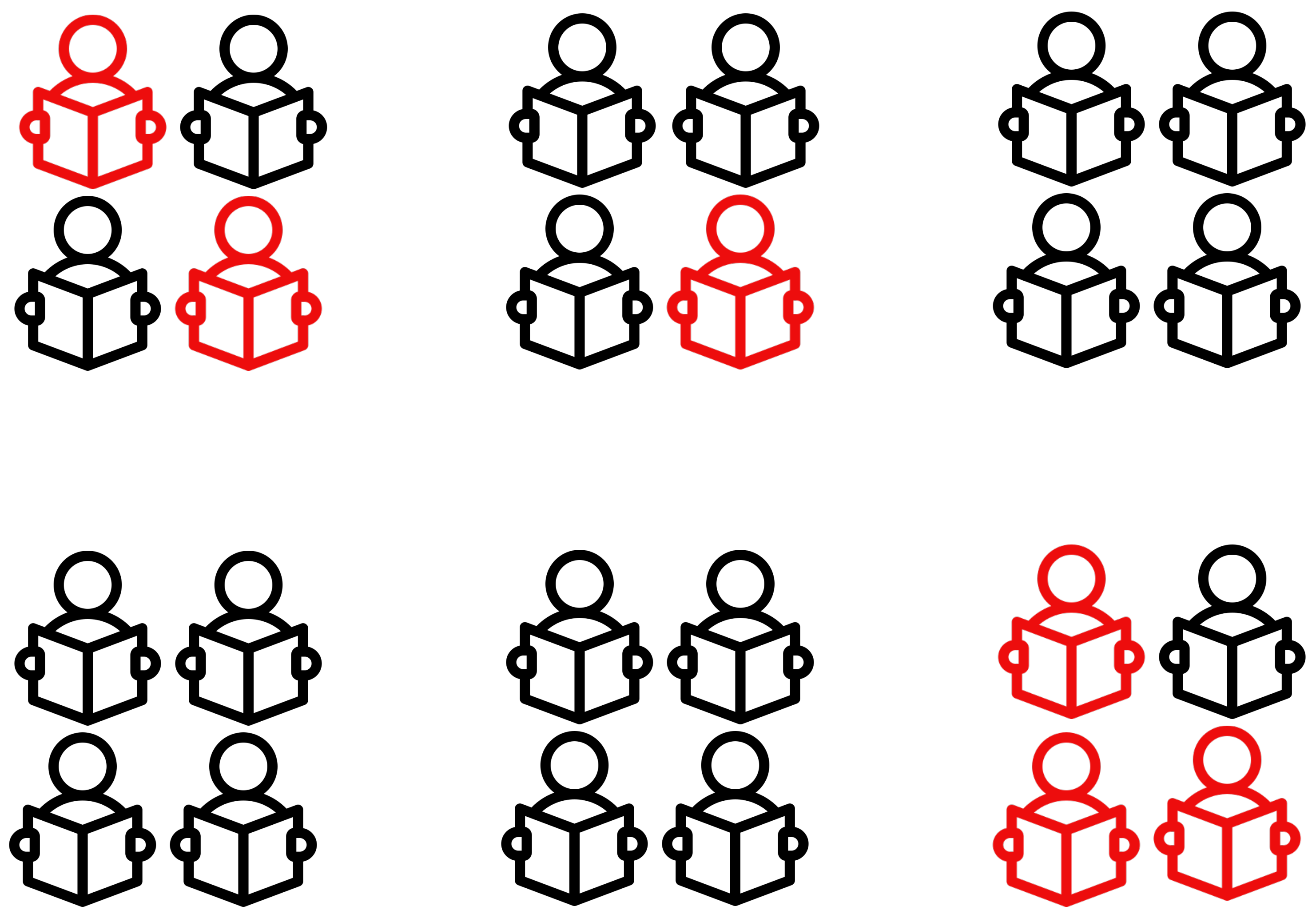
What if more classrooms are infected?



- Propose some constructions of $(\mathscr{S}, r) - CFF$
  - For m classrooms of k students each
  - Identifies r infected classrooms and everyone inside them

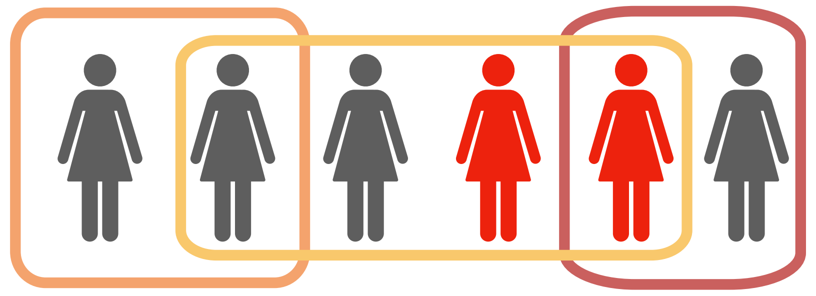
Kronecker-type construction

What if more classrooms are infected?



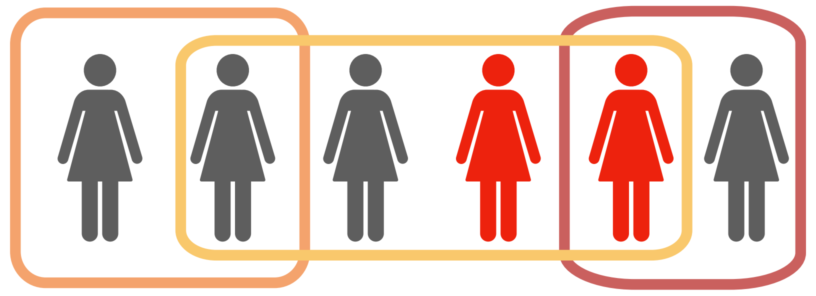
- Propose some constructions of $(\mathcal{S}, r) - CFF$
  - For m classrooms of k students each
  - Identifies r infected classrooms and everyone inside them
- Generalization of Li, van Rees and Wei (2006)
  - Uses an $r - CFF(t, m)$ to build $(\mathcal{S}, r) - ECFF(t, km)$ and $(\mathcal{S}, r) - CFF(kt, km)$
- Allows edges of different cardinalities

Overlapping edges



- Explore the properties of the hypergraph
- Propose constructions inspired by the non-overlapping ones
  - Construction of $(\mathscr{S},1) - CFF$ and $(\mathscr{S},1) - ECFF$ based on edge-colouring

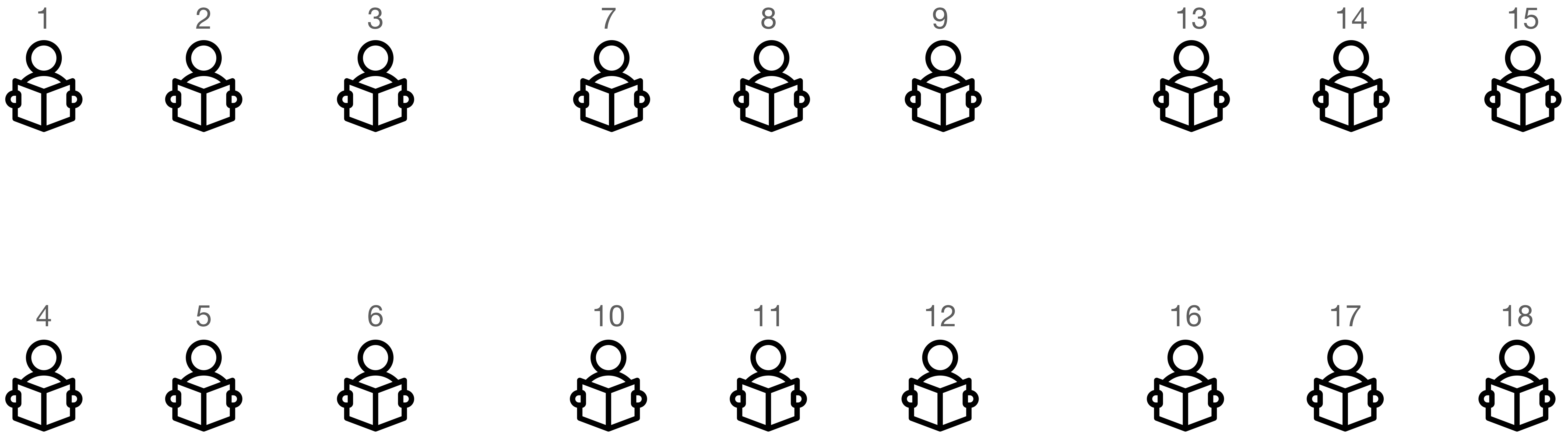
Overlapping edges



- Explore the properties of the hypergraph
- Propose constructions inspired by the non-overlapping ones
  - Construction of $(\mathcal{S},1) - CFF$ and $(\mathcal{S},1) - ECFF$ based on edge-colouring
  - Construction of $(\mathcal{S},r) - CFF$ based on strong edge-colouring
    - ***Defect cover***: a set of at most r edges whose union contains the set of infected elements
    - We can handle many infected edges, as long as the size of the defect cover is $\leq r$

The high school problem

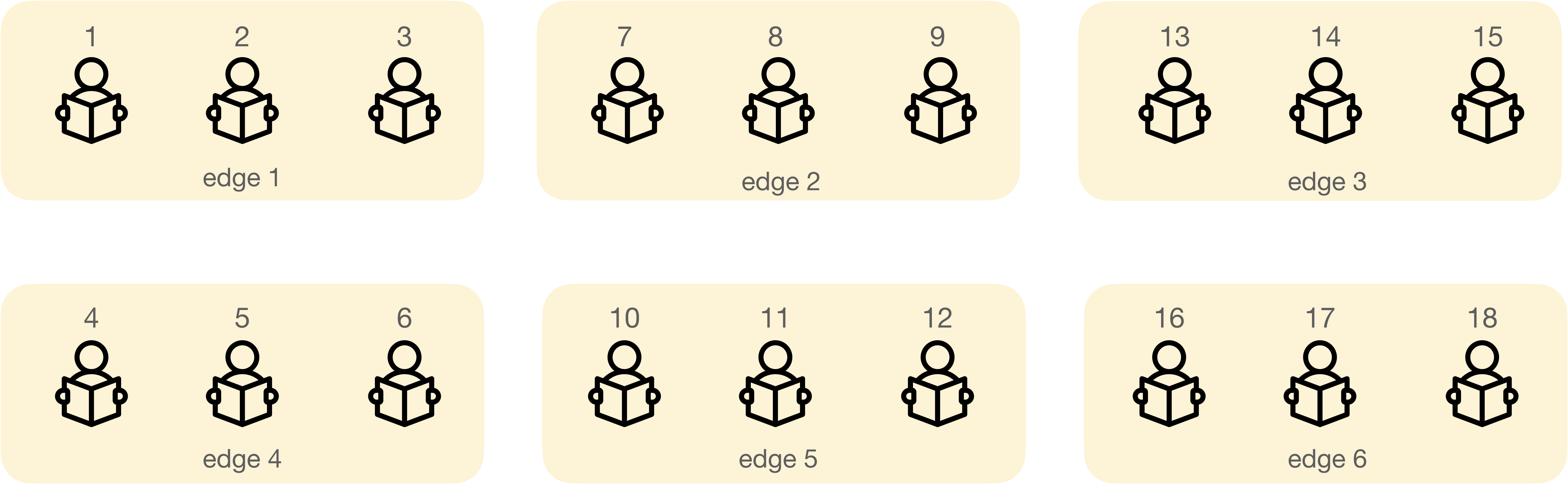
Construction



The high school problem

Construction

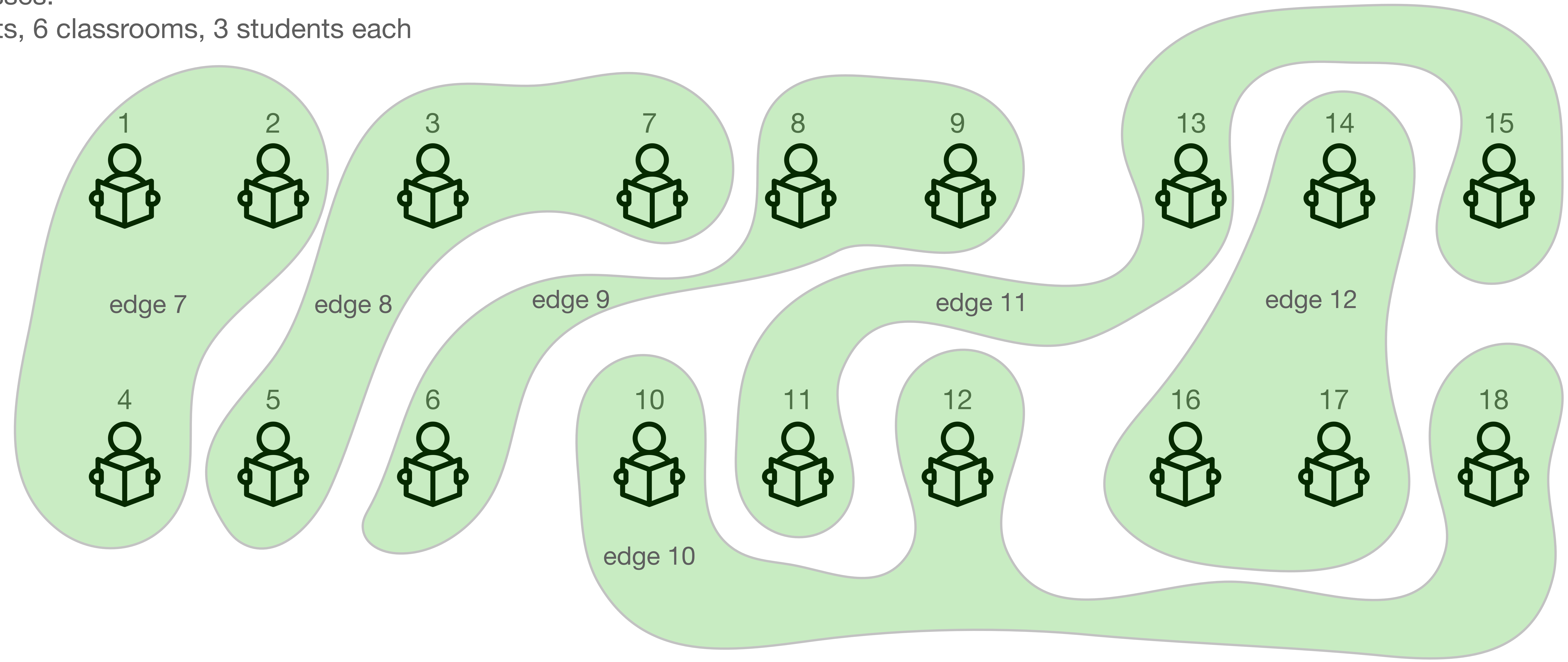
Morning classes:
n = 18 students, 6 classrooms, 3 students each



The high school problem

Construction

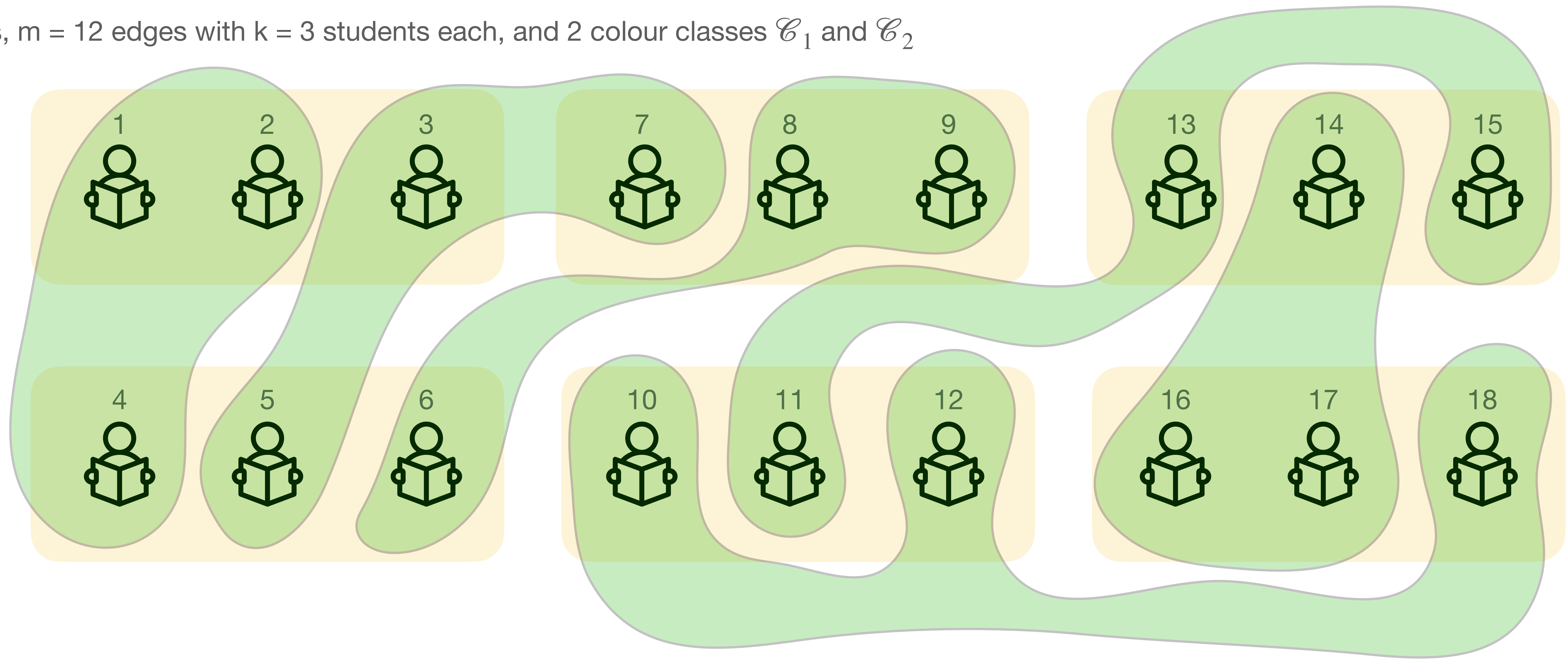
Afternoon classes:
n = 18 students, 6 classrooms, 3 students each



The high school problem

Construction

Total:
n = 18 vertices, m = 12 edges with k = 3 students each, and 2 colour classes $\mathscr{C}_1$ and $\mathscr{C}_2$



Overlapping edge construction

$\mathscr{C}_1$

edge 1	edge 2	edge 3	edge 4	edge 5	edge 6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

$\mathscr{C}_2$

edge 7	edge 8	edge 9	edge 10	edge 11	edge 12
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0

Overlapping edge construction

$\mathscr{C}_1$

edge 1	edge 2	edge 3	edge 4	edge 5	edge 6
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

$\mathscr{C}_2$

edge 7	edge 8	edge 9	edge 10	edge 11	edge 12
1	1	1	0	0	0
1	0	0	1	1	0
0	1	0	1	0	1
0	0	1	0	1	1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

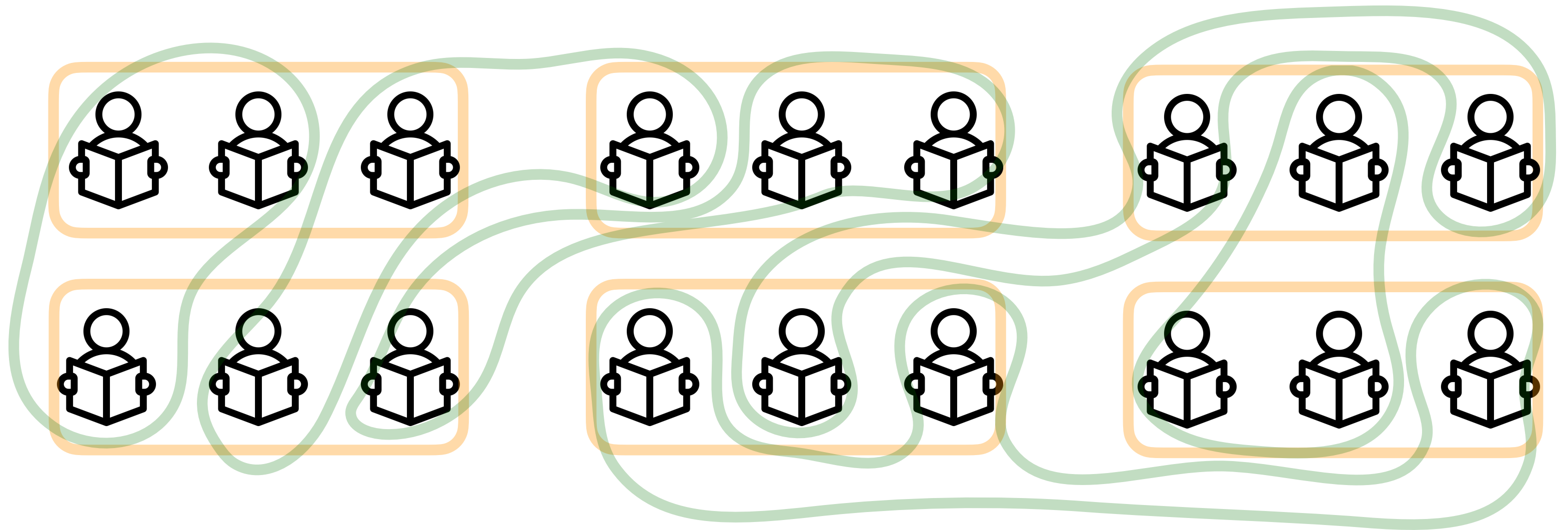
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

Overlapping edge construction

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$$(\mathcal{S},1) - CFF(t,n)$$



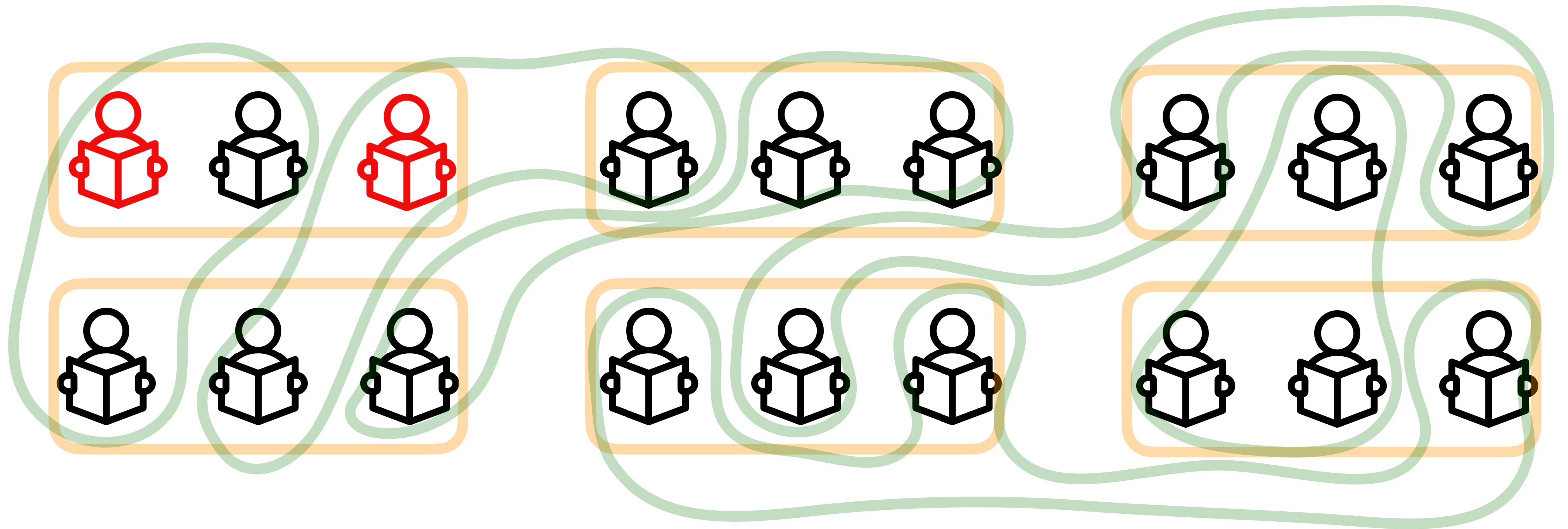
Overlapping edge construction

Edges 1, 7 and 8 are infected

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$(\mathcal{S},1) - CFF(t,n)$



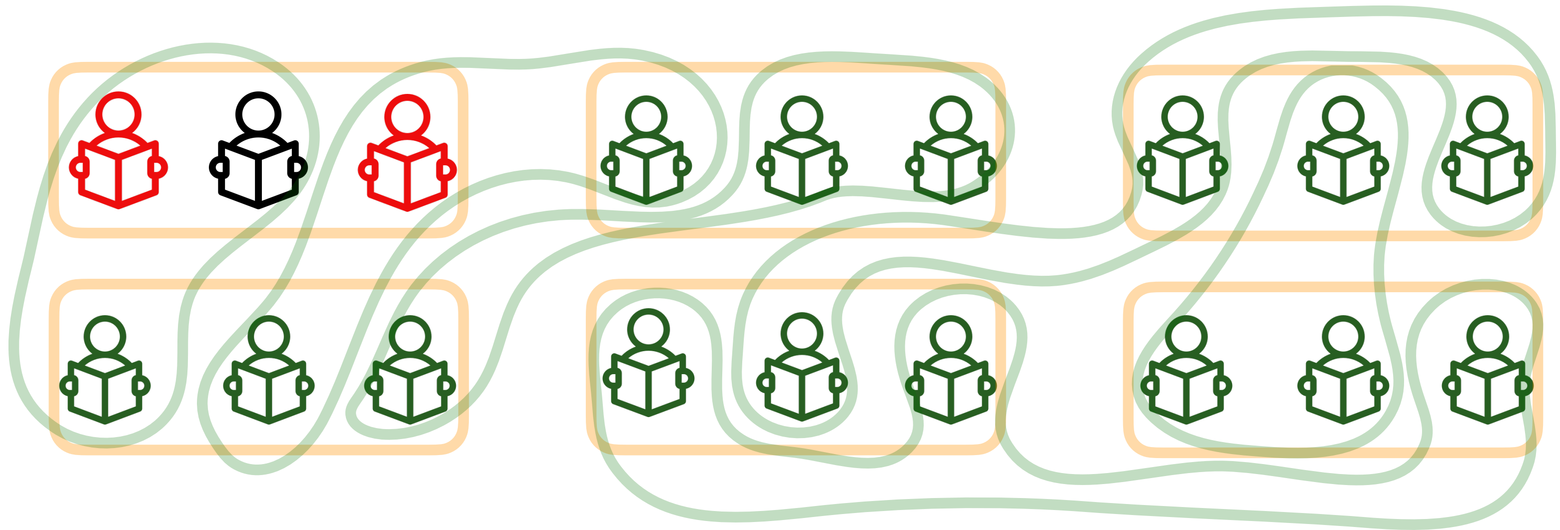
Overlapping edge construction

Edges 1, 7 and 8 are infected, students 4-18 are cleared out

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0
1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

$$(\mathcal{S},1) - CFF(t,n)$$



Overlapping edge construction

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1

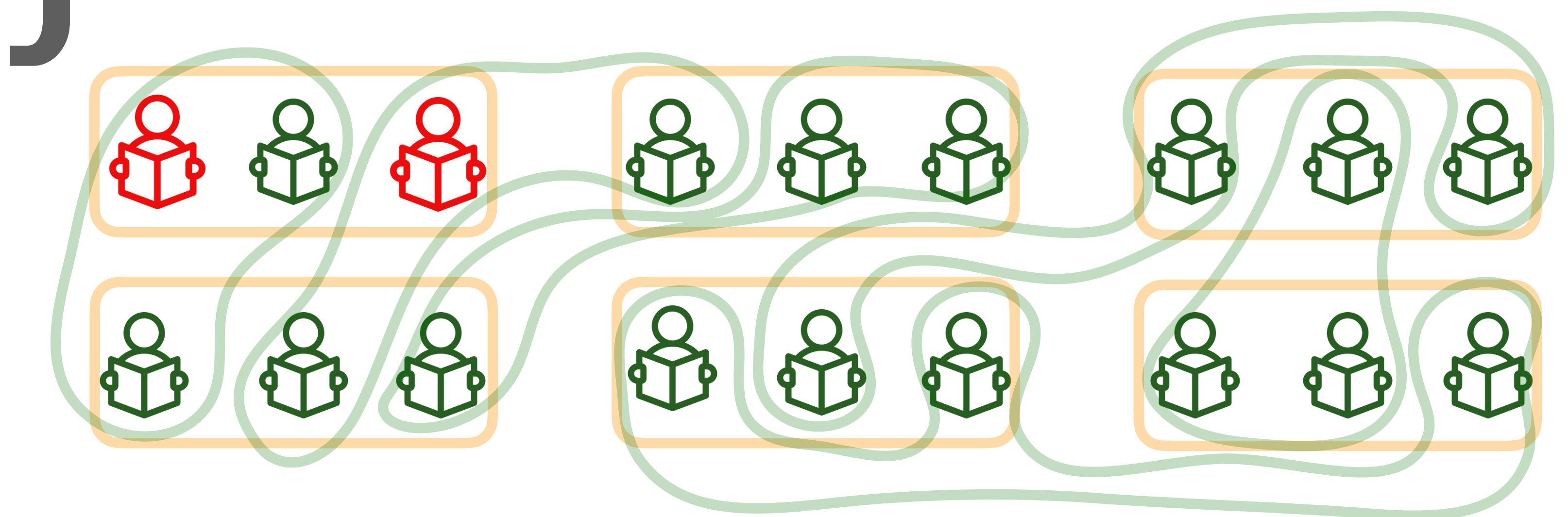
1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1

Edges 1, 7 and 8 are infected, students 4-18 are cleared out

Inside each infected edge, we can identify precisely the infected students

1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	1	1
0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	1	0

1	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0
0	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1

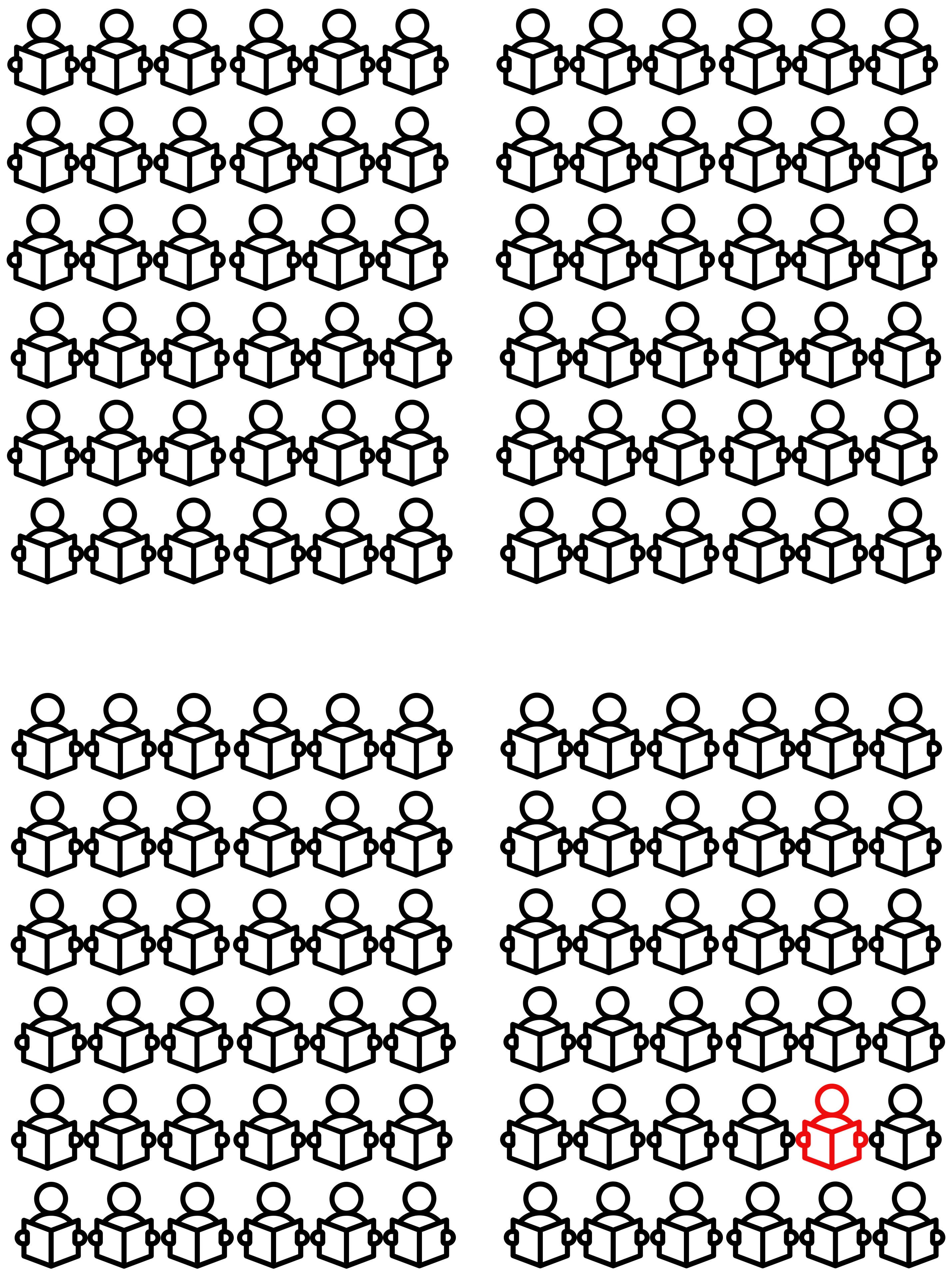


$$(\mathcal{S},1) - CFF(t = 2 \times (4+3), n = 18)$$

$$(\mathcal{S},1) - ECFF(t' = 2 \times 4, n = 18)$$

For a larger high school

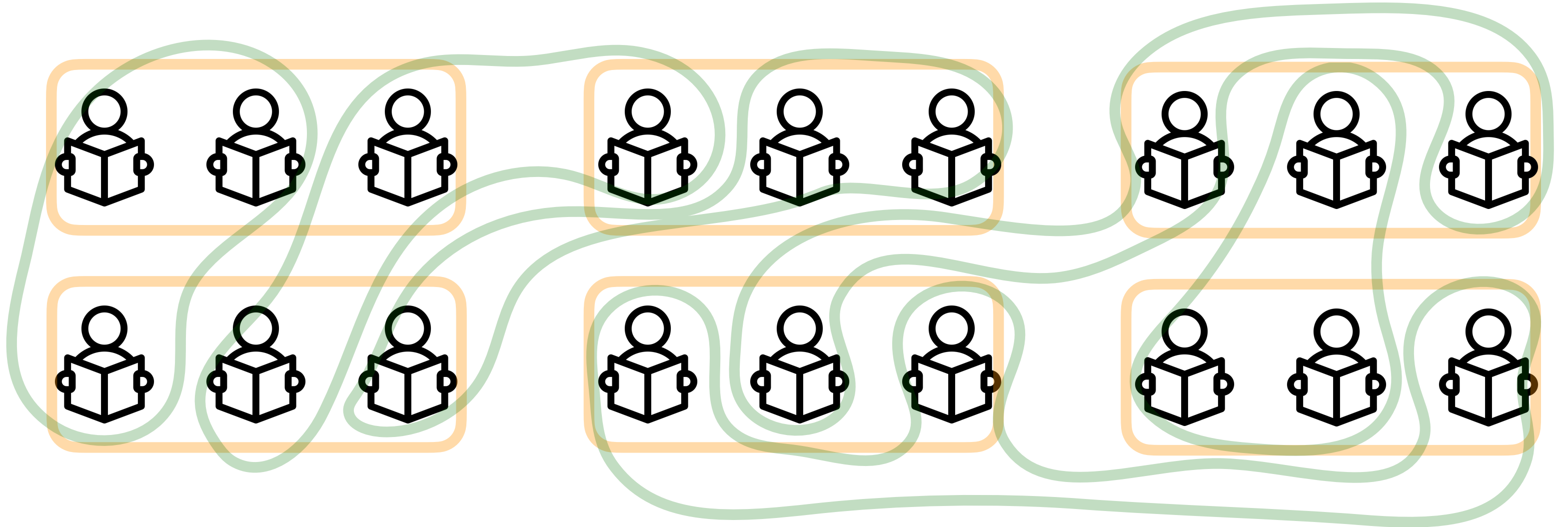
- **n = 900** studens
- Each student taking 4 courses (4 colour classes)
- Total of m = 120 courses (edges)
- Each course with 30 students (cardinality of edges)
- Tests:
  - Use $1 - CFF(7{,}30 = 120/4)$
  - t’ = 7x4 = **28** tests to detect infected edges (course of outbreak)
  - t = 28+30x4 = **148** tests to identify all infected individuals



Overlapping edge construction

$(\mathcal{S},1) - CFF(t,n)$

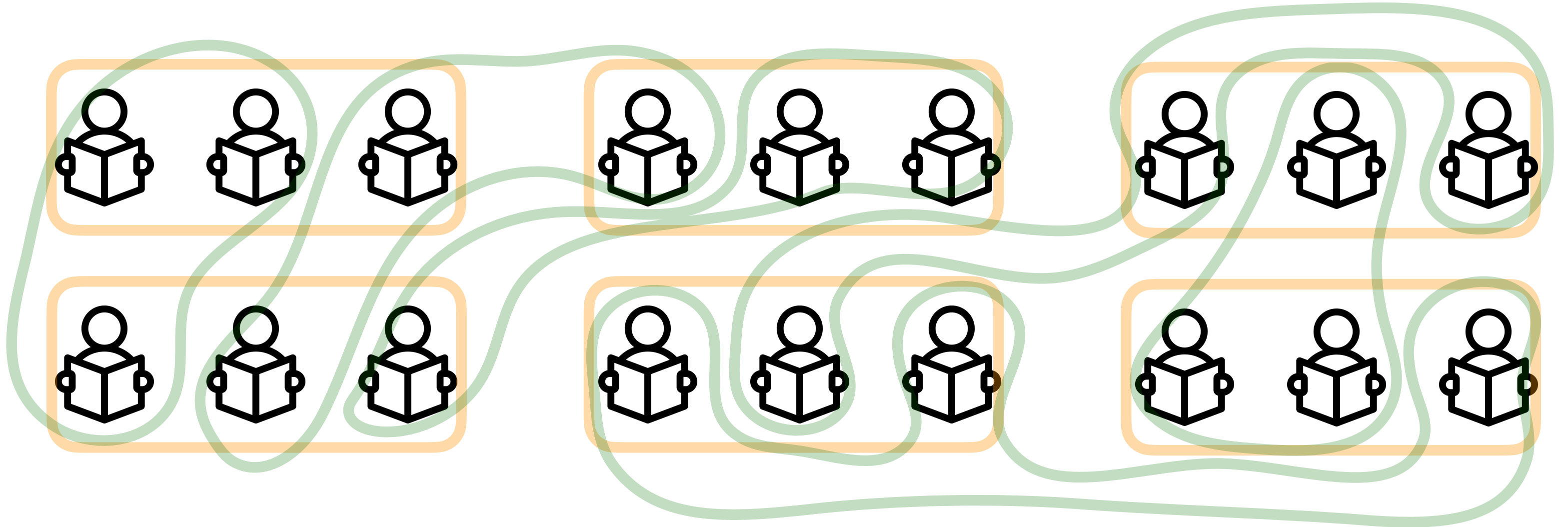
- Consider a hypergraph $\mathcal{H}$ with edge chromatic number $\chi(\mathcal{H}) = \ell$ and colour classes $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$
- If $\mathcal{H}$ is **k-uniform**: we have $(\mathcal{S},1) - CFF(t,n)$ and $(\mathcal{S},1) - ECFF(t',n)$
  - Start with a $1 - CFF(t_1, n/k)$
  - $t \leq \ell \times (t_1 + k) \approx \ell \times (\log n/k + k)$
  - $t' \leq \ell \times t_1 \approx \ell \times \log n/k$



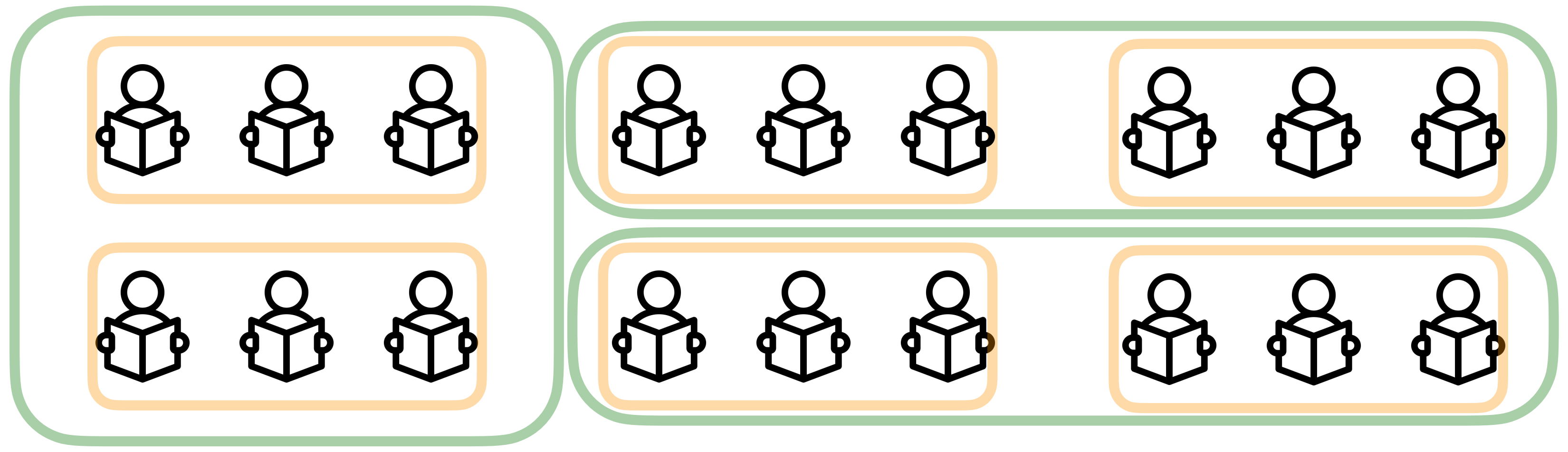
Overlapping edge construction

$(\mathcal{S},1) - CFF(t,n)$

- Consider a hypergraph $\mathcal{H}$ with edge chromatic number $\chi(\mathcal{H}) = \ell$ and colour classes $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$
- If $\mathcal{H}$ is **k-uniform**: we have $(\mathcal{S},1) - CFF(t,n)$ and $(\mathcal{S},1) - ECFF(t',n)$
  - Start with a $1 - CFF(t_1, n/k)$
  - $t \leq \ell \times (t_1 + k) \approx \ell \times (\log n/k + k)$
  - $t' \leq \ell \times t_1 \approx \ell \times \log n/k$



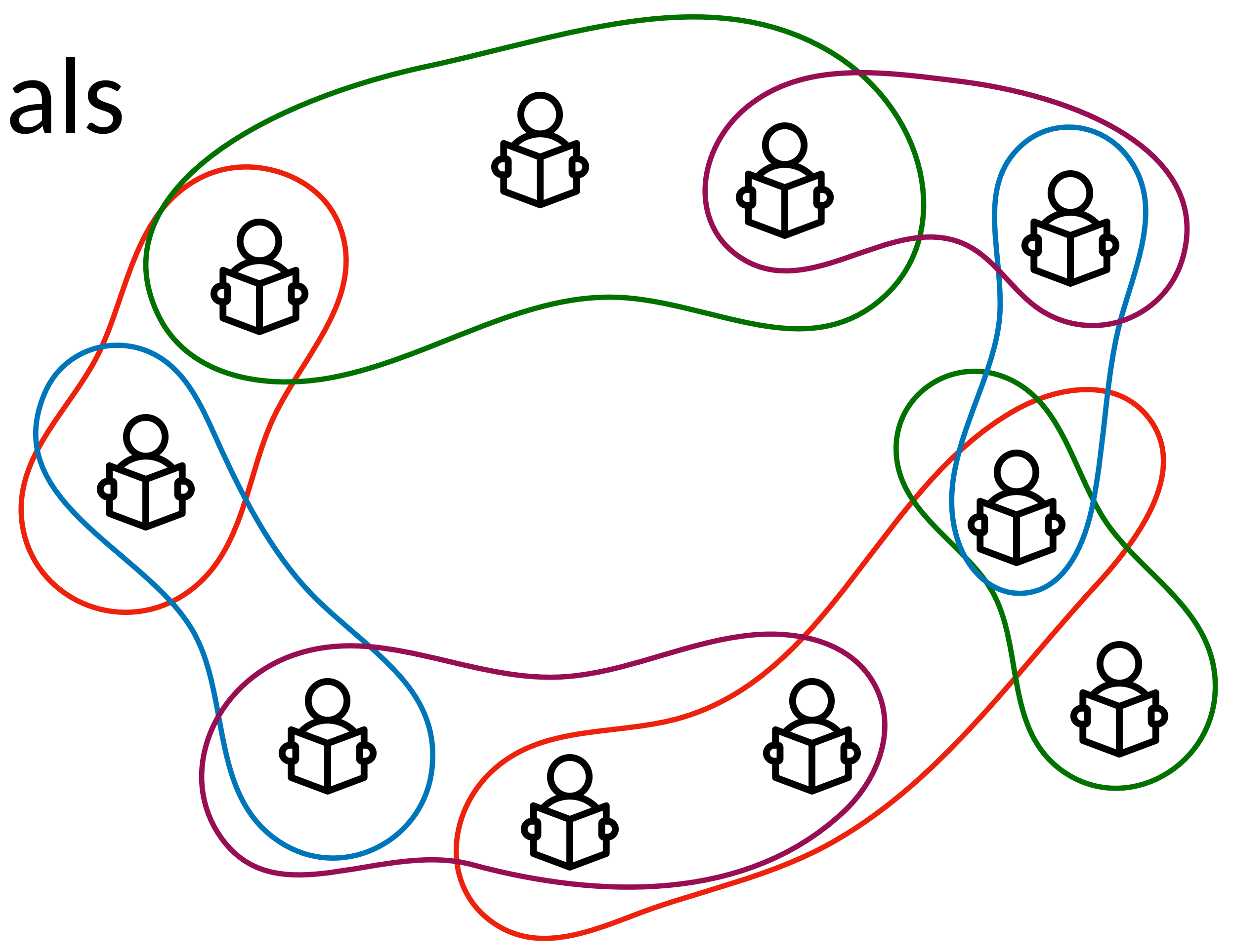
- If $\mathcal{H}$ has edges of **different cardinalities**, we have $(\mathcal{S},1) - CFF(t,n)$ and $(\mathcal{S},1) - ECFF(t',n)$
  - Start with $1 - CFF(t_i, |\mathcal{C}_i| + \delta_i), 1 \leq i \leq \ell$
  - $t = \sum_{i=1}^{\ell} (t_i + k_i), \quad k_i =$ max edge in colour class $\mathcal{C}_i$
  - $t' = \sum_{i=1}^{\ell} t_i$



Overlapping edge construction

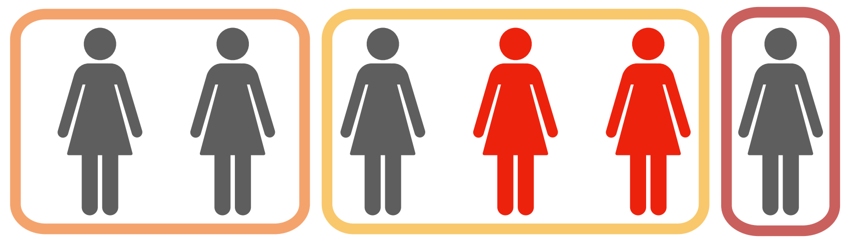
$(\mathcal{S}, r) - CFF(t, n)$

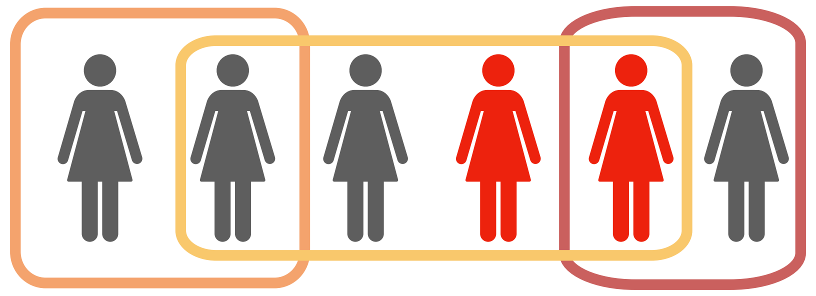
- Generalization for $(\mathcal{S}, r) - CFF(t, n)$ using **strong edge-colouring**
  - Assuming that r edges $\mathcal{E} = \{S_1, S_2, \ldots, S_r\}$ contain all infected individuals
  - There are at most r edges in $\mathcal{C}_i$ which intersect $\mathcal{E}$
    - $\mathcal{C}_i$ contains at most r infected edges
  - Use a combination of $r - CFF(t_i, |\mathcal{C}_i|)$ and $(r-1) - CFF(t_i', |\mathcal{C}_i|)$
  - $(\mathcal{S}, r) - CFF(t, n)$ with $t \leq \sum_{i=1}^{\ell} (t_i + k_i t_i')$, k_i = max edge in colour class $\mathcal{C}_i$



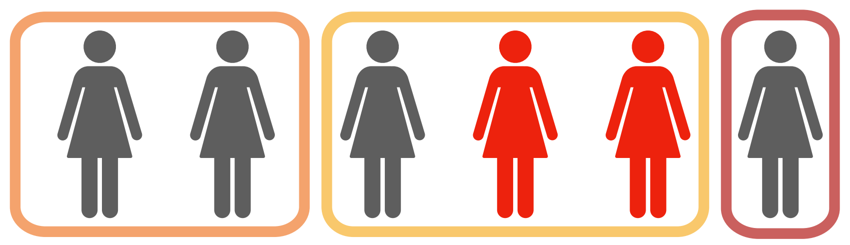
Structure-aware CFFs

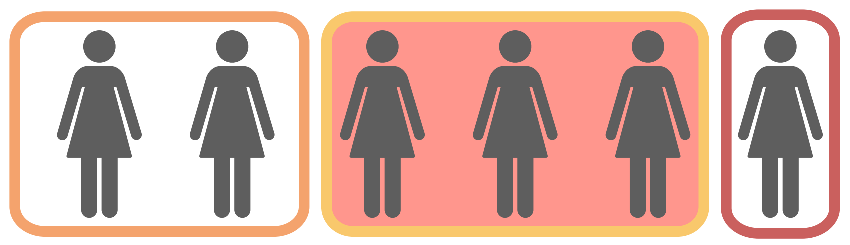
Overlapping and non-overlapping edges:





Configurations: $(\mathscr{S}, r) - CFF(t, n)$ and $(\mathscr{S}, r) - ECFF(t, n)$

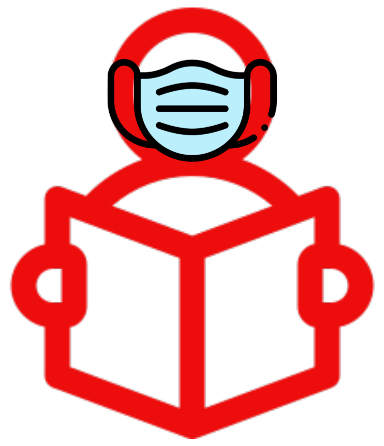




Future work on structure-aware CFFs

- Explore other constraints of the applications
  - Limit on number of 1s per row and/or column
- Generalize definitions to allow flexible internal identification
  - Assume a bound on the number of infected items inside an edge (instead of edge size)
- Explore probabilistic constructions
- Compare constructions with known lower bounds

Thank you!



Thais Bardini Idalino - thais.bardini@ufsc.br
Lucia Moura - lmoura@uottawa.ca

uOttawa UFSC