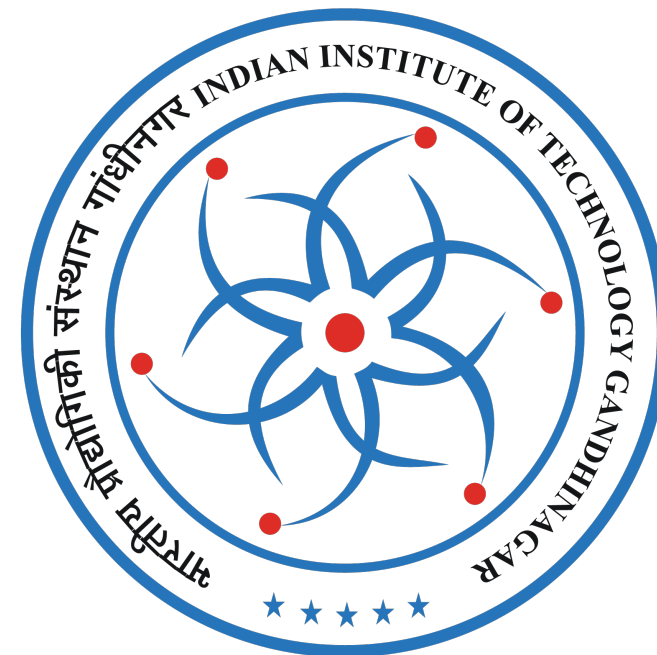


Deleting to Structured Trees

Pratyush Dayal and Neeldhara Misra
IIT Gandhinagar



COCOON 2019
XIAN, CHINA — 29TH JULY

Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

Once upon a time*, there was the FEEDBACK VERTEX SET problem.

FEEDBACK VERTEX SET

Forest

*Classical NP-complete problem

Related question (1): delete vertices to obtain a *tree*.

TREE DELETION SET

Tree

Related question (2): delete vertices to obtain a *forest of pathwidth one*.

PW-ONE DELETION SET

Caterpillars

Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk.

An improved FPT algorithm and a quadratic kernel for pathwidth one vertex deletion. *Algorithmica*, 64(1):170–188, 2012.

Geevarghese Philip, Venkatesh Raman, and Yngve Villanger. A quartic kernel for pathwidth-one vertex deletion. (WG 2010)

Related question (3): delete vertices to obtain *stars of bounded degree*.

STAR DELETION SET

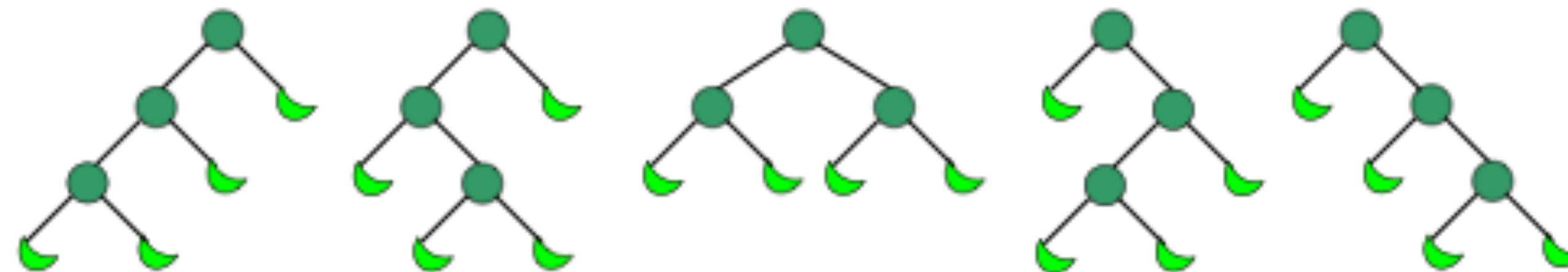
Stars

Our problem: delete vertices to obtain a **full binary tree**.

FBT DELETION SET

*Full
binary tree*

Rooted full binary tree: every vertex has either two children or no children.



Phylogenetic trees, polymerization processes, etc.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and interesting property that is hereditary on induced subgraphs.

The problem of finding a maximum connected subgraph satisfying a **property π** is NP-hard for any non-trivial and interesting property that is hereditary on induced subgraphs.

A property is a class of graphs.

We will say that the property is satisfied by, or is true for, a graph G if G belongs to the class given by the property.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any **non-trivial** and interesting property that is hereditary on induced subgraphs.

A property is said to be non-trivial if it is true for at least one graph and false for at least one graph.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and **interesting** property that is hereditary on induced subgraphs.

A property is said to be interesting if it is true for arbitrarily large graphs.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and interesting property that is **hereditary on induced subgraphs**.

A property is said to be hereditary on induced subgraphs if the deletion of any node from a graph that has the property always results in a graph that also has the property.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and interesting property that is hereditary on induced subgraphs.

TREE DELETION SET

Plug in the property of acyclicity above.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and interesting property that is hereditary on induced subgraphs.

FBT DELETION SET

The property of being a binary tree is not hereditary on induced subgraphs.

The problem of finding a maximum connected subgraph satisfying a property π is NP-hard for any non-trivial and interesting property that is hereditary on induced subgraphs.

FBT DELETION SET

The property of being a binary tree is not hereditary on induced subgraphs.



Our Contributions

Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Deleting optimally to full binary trees by removing *edges* is NP-complete.

Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Deleting optimally to full binary trees by removing edges is NP-complete.



Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Deleting optimally to full binary trees by removing *edges* is NP-complete.



Deleting optimally to *trees* by removing edges is polynomial time (complement of a MST).

Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Deleting optimally to full binary trees by removing *edges* is NP-complete.

Our Contributions

Deleting optimally to full binary trees by removing vertices is NP-complete.

Deleting optimally to full binary trees by removing *edges* is NP-complete.

Both problems are FPT when parameterized by solution size.

Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

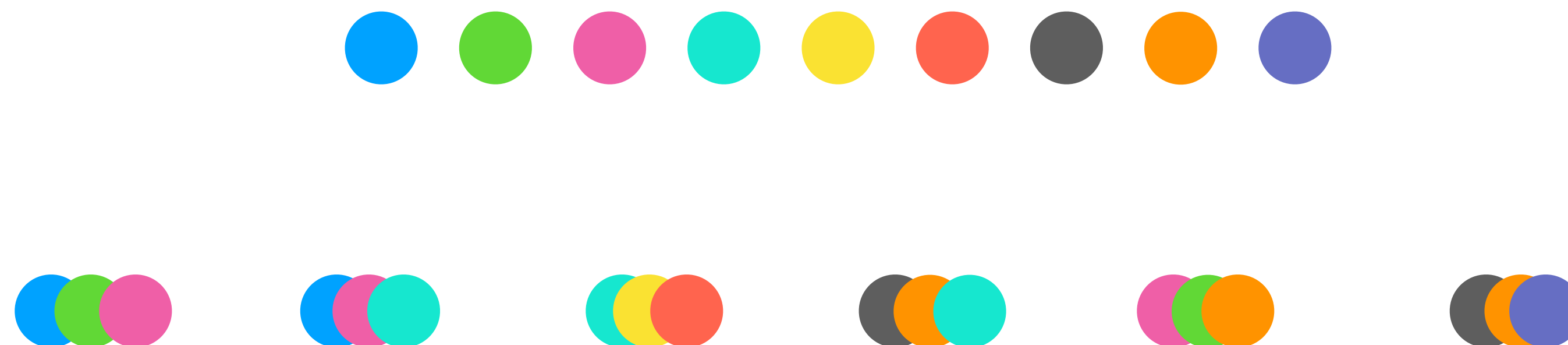
Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

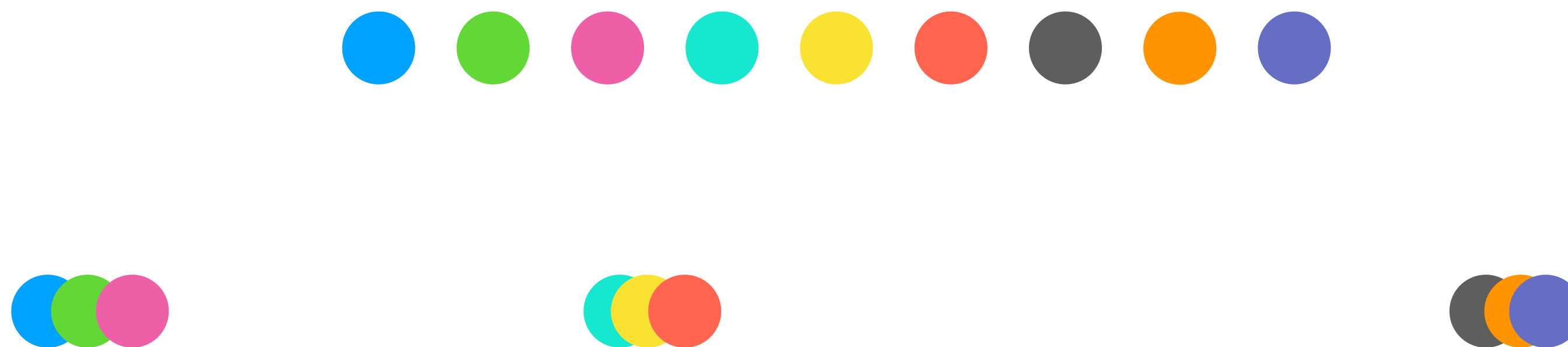
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



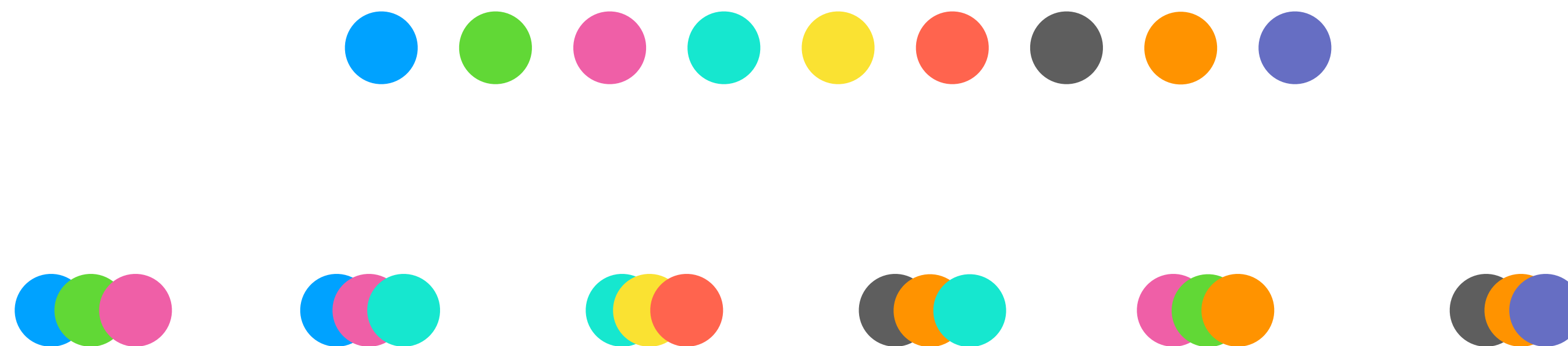
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



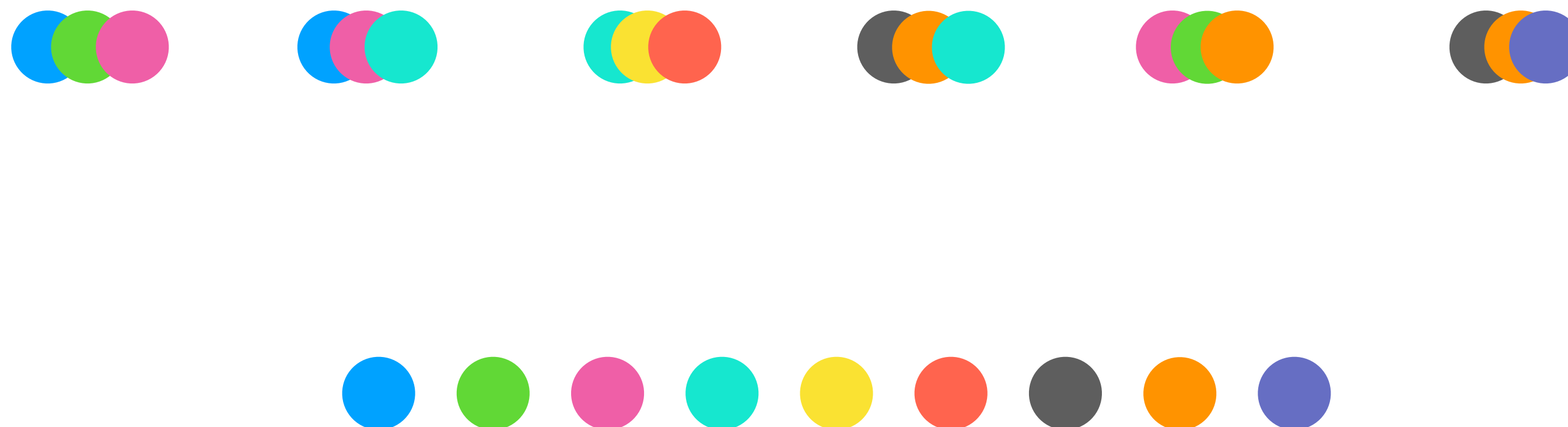
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



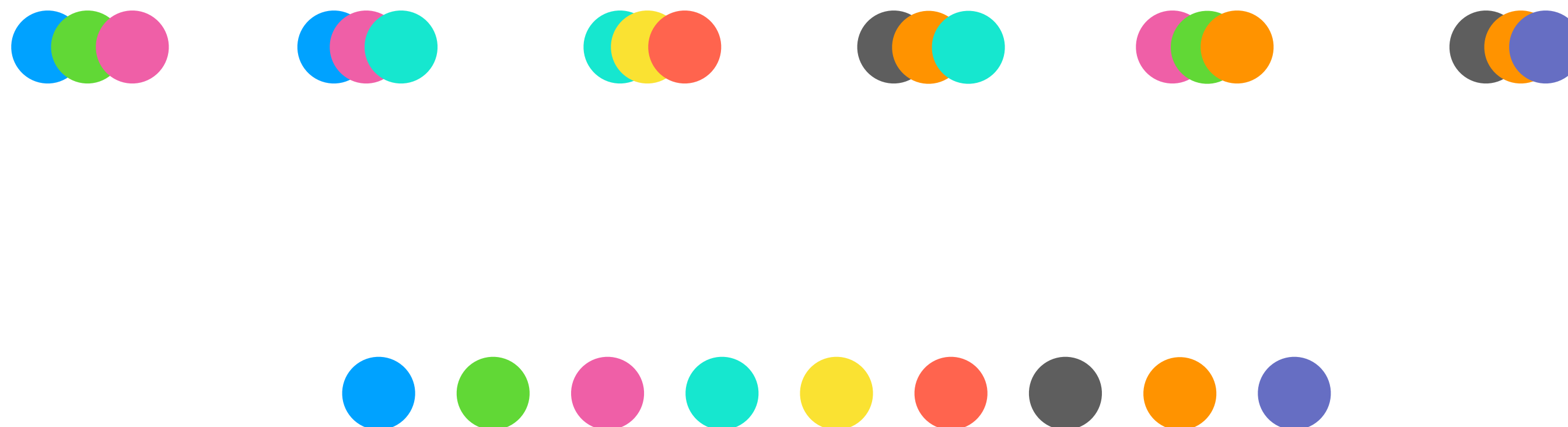
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



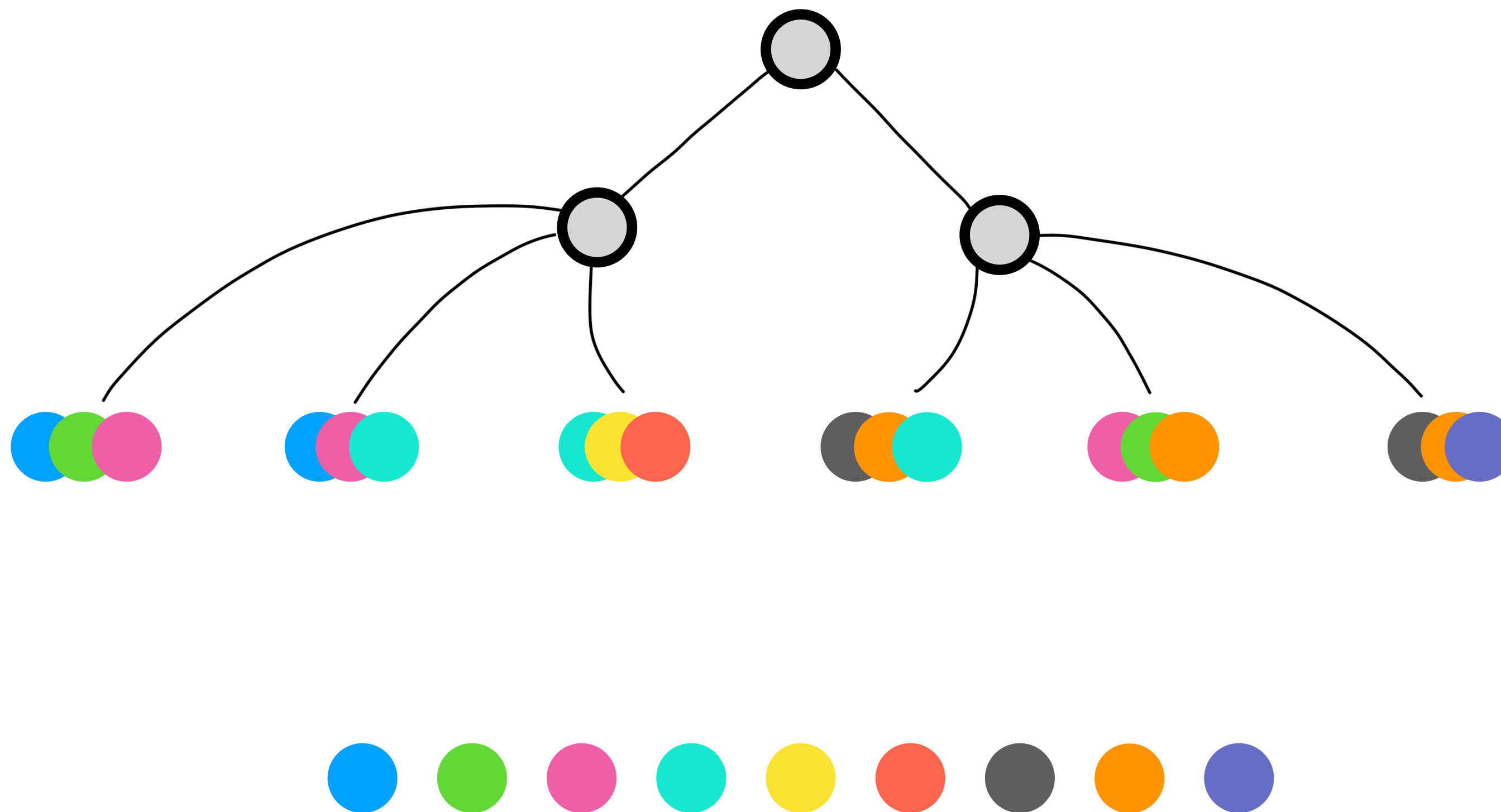
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



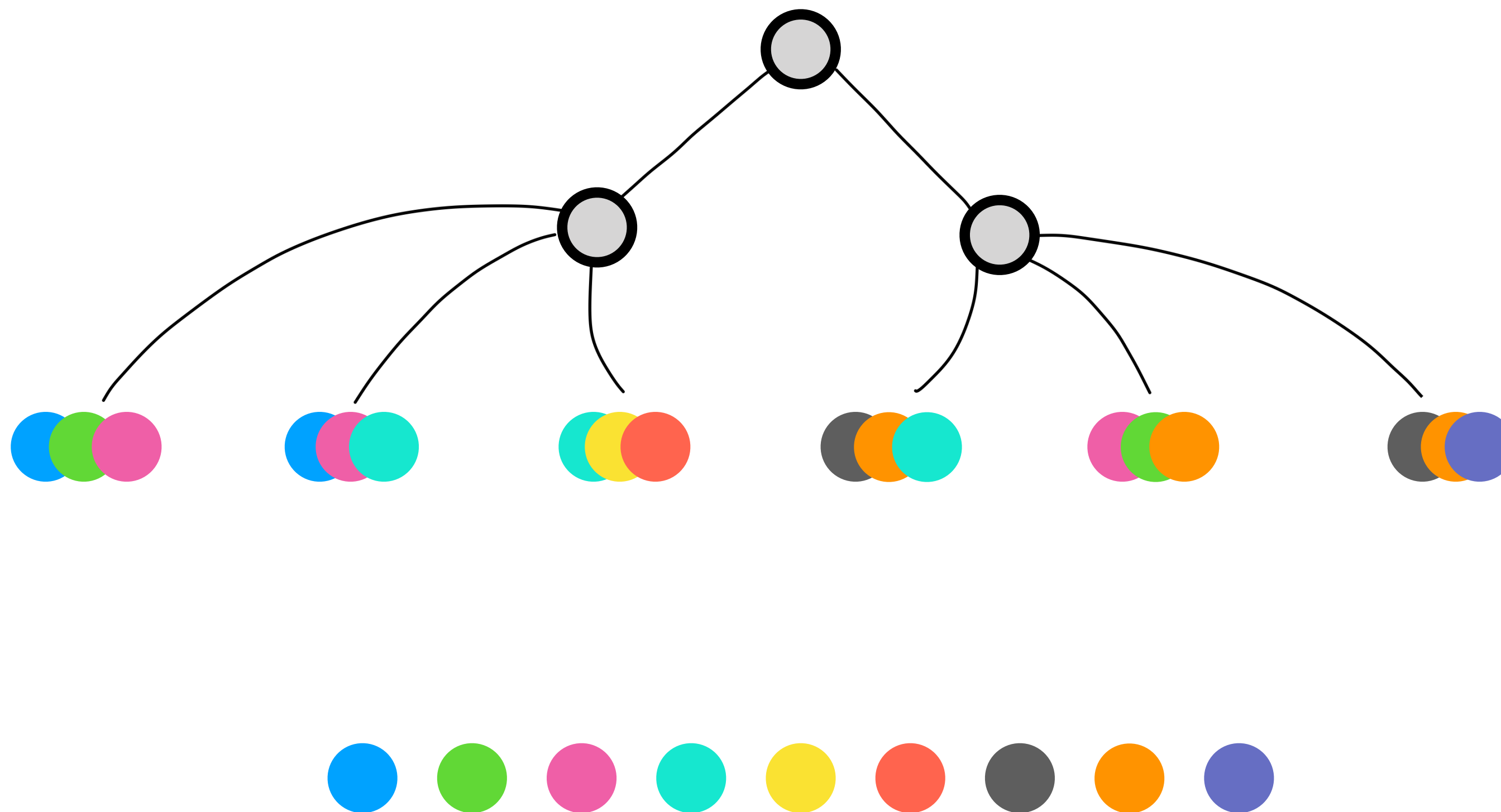
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



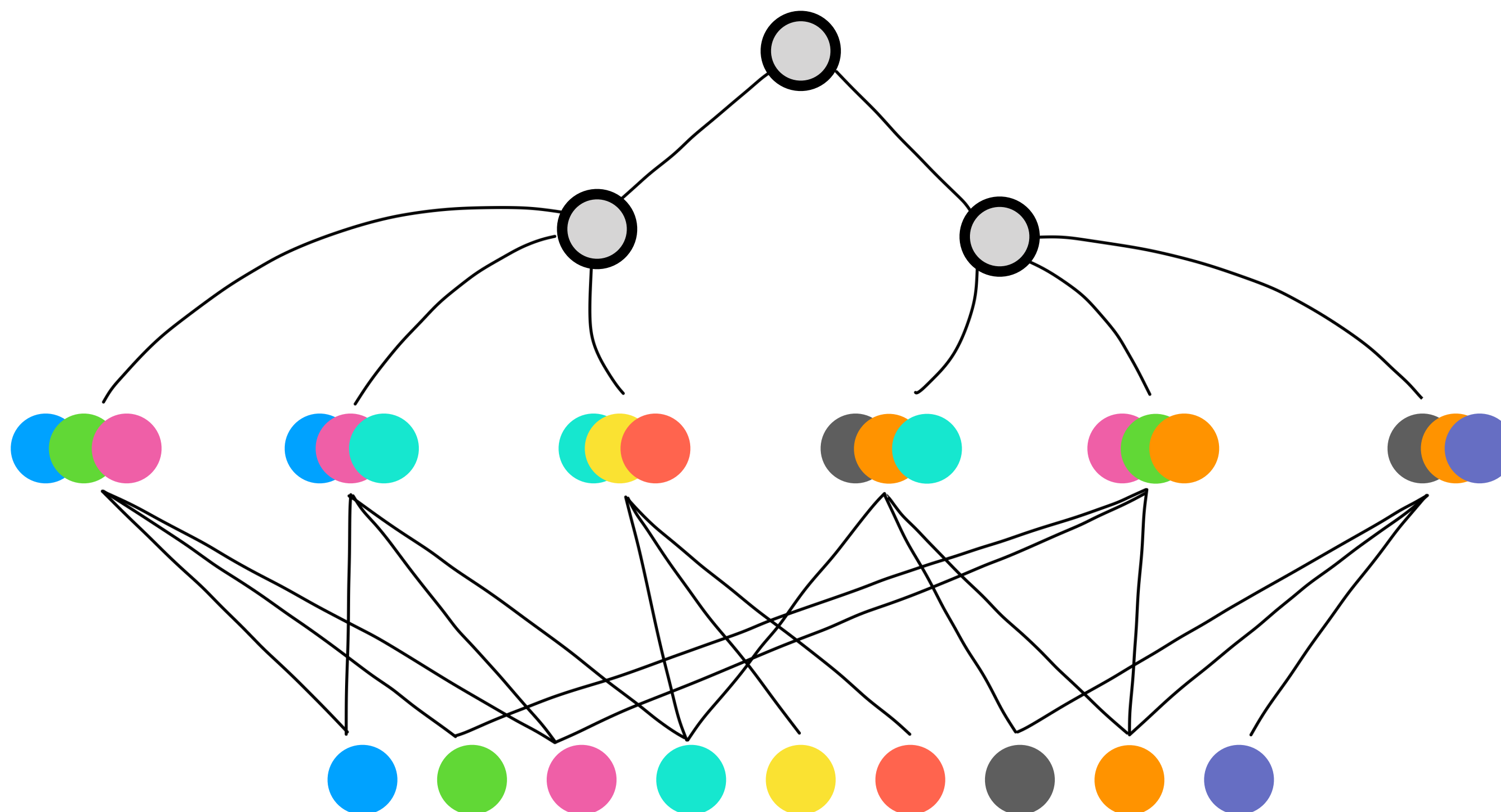
Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



Deleting to ternary trees

Reduce from EXACT COVER BY 3-SETS



Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

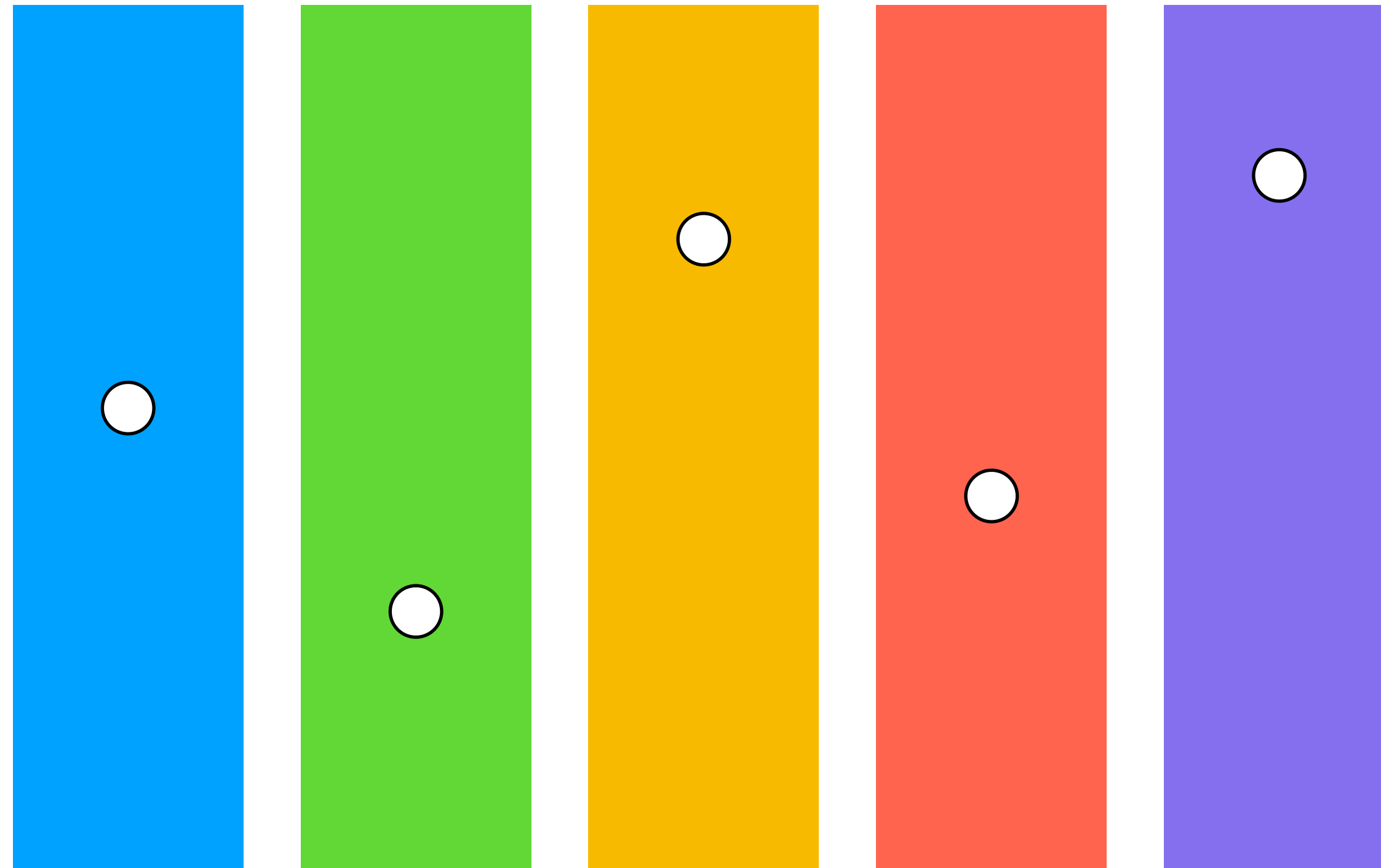
Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

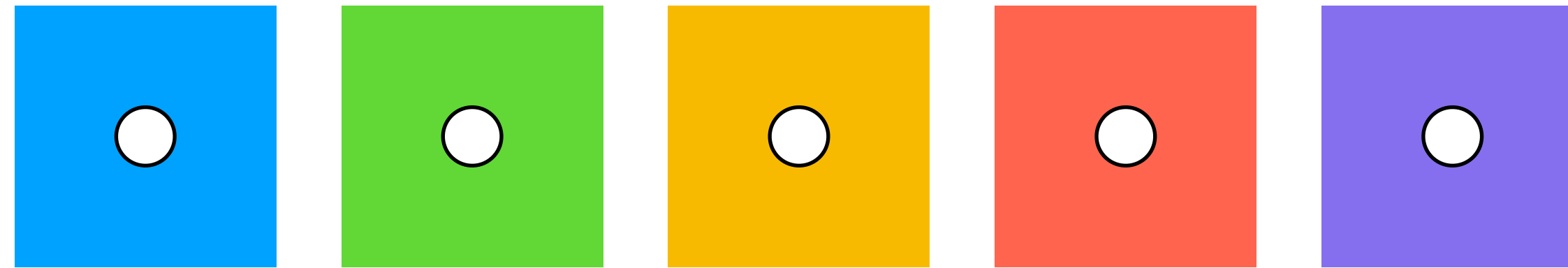
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



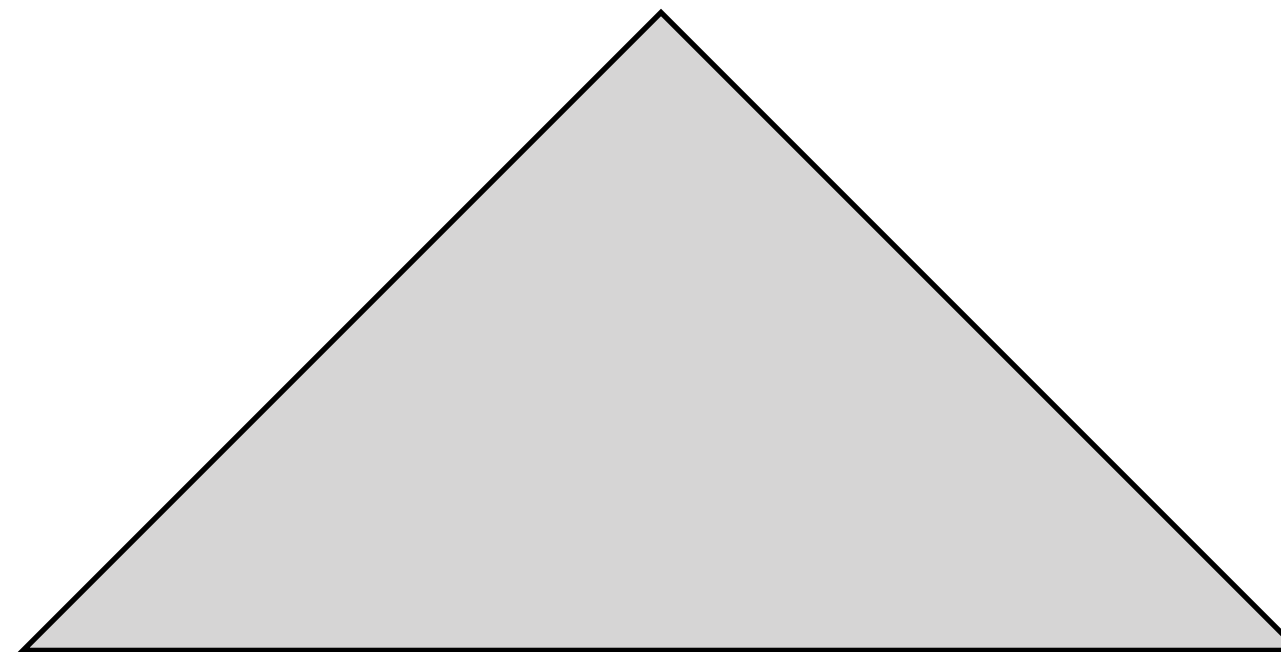
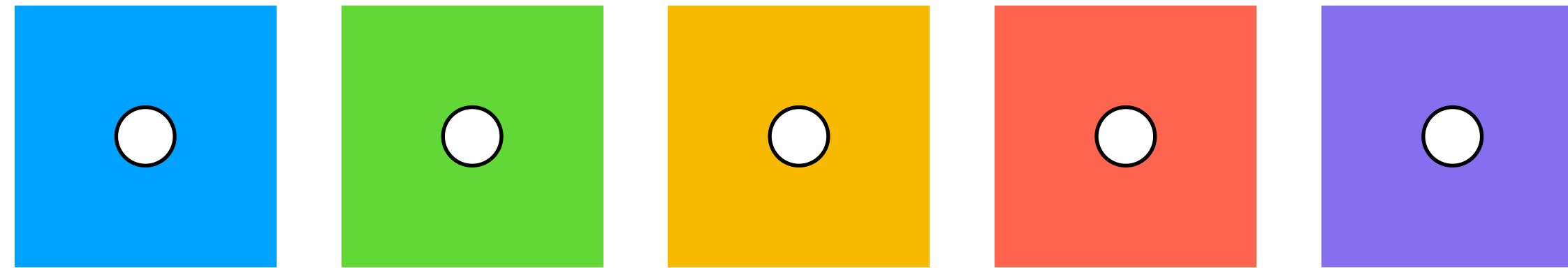
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



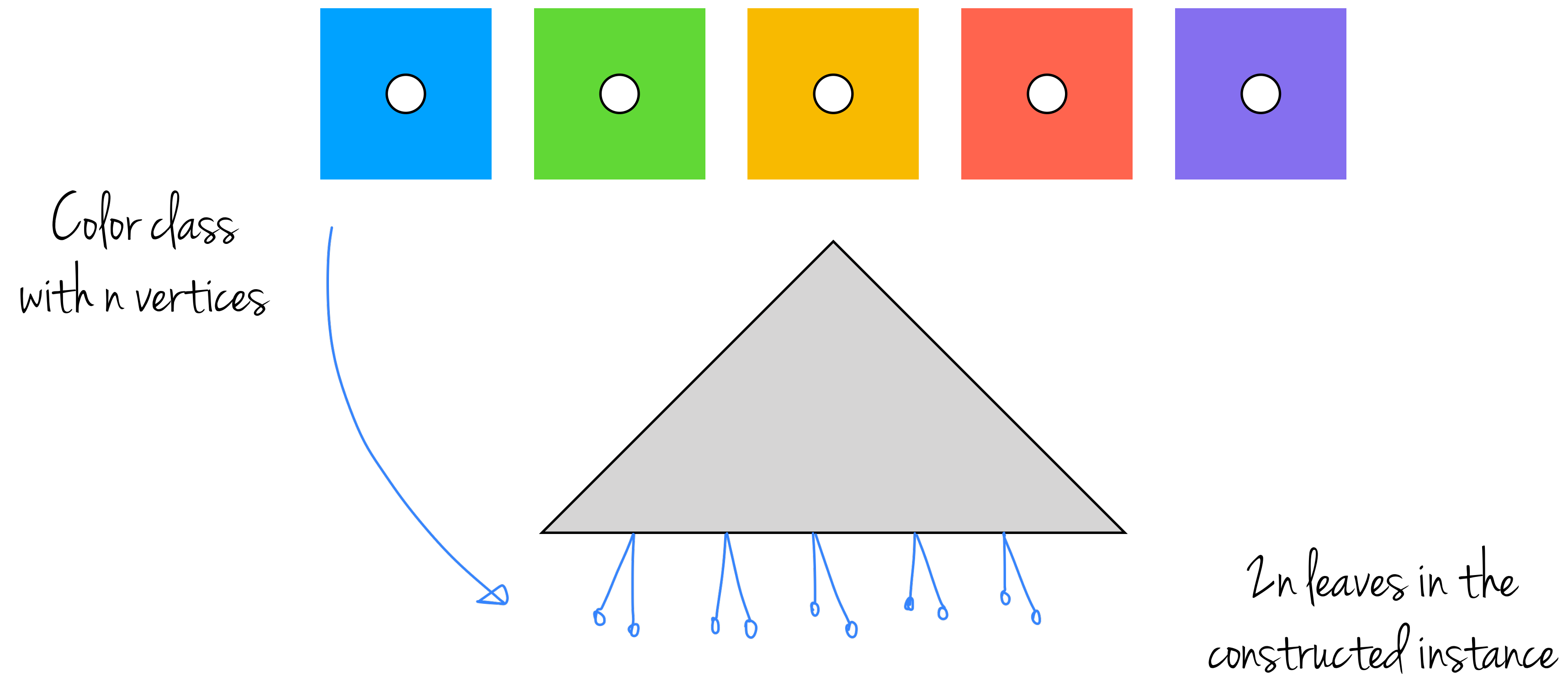
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



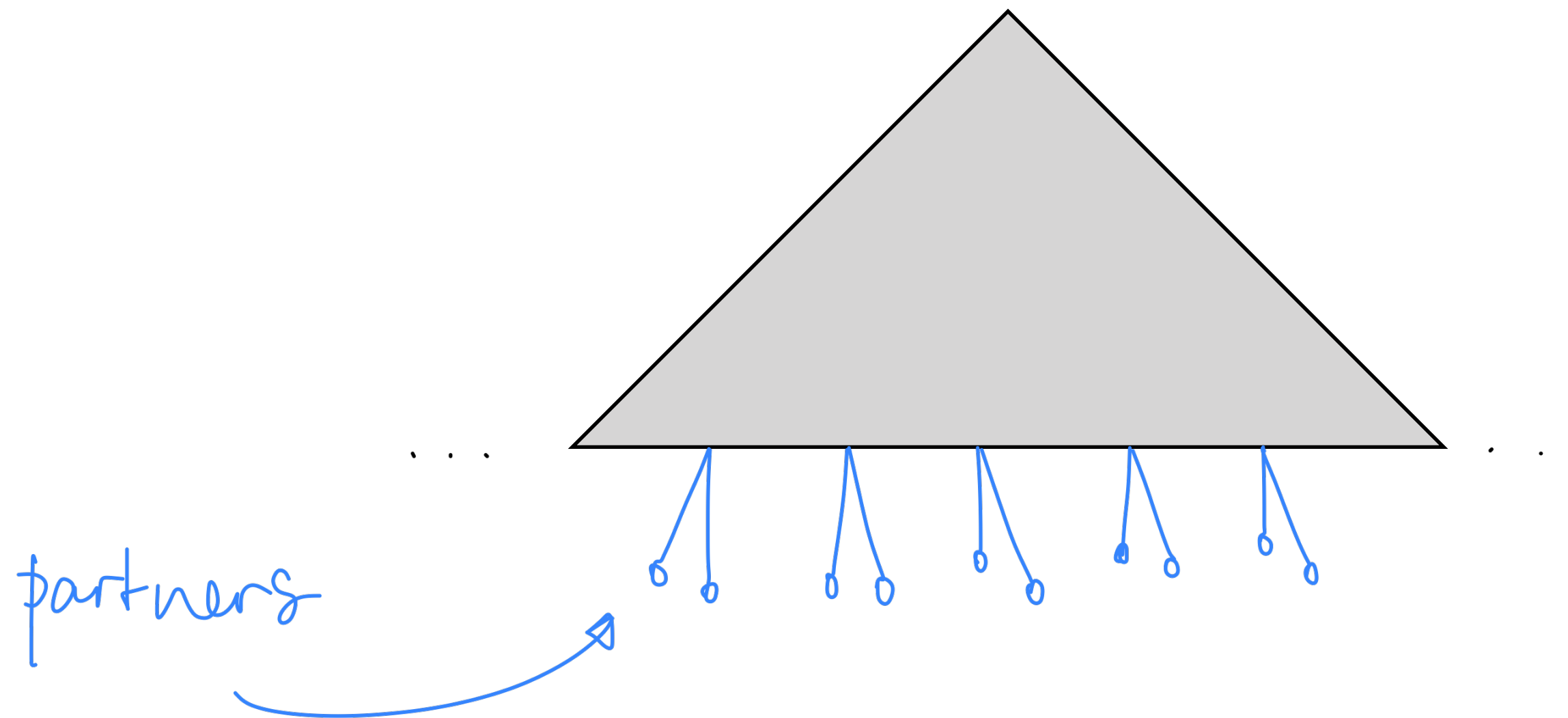
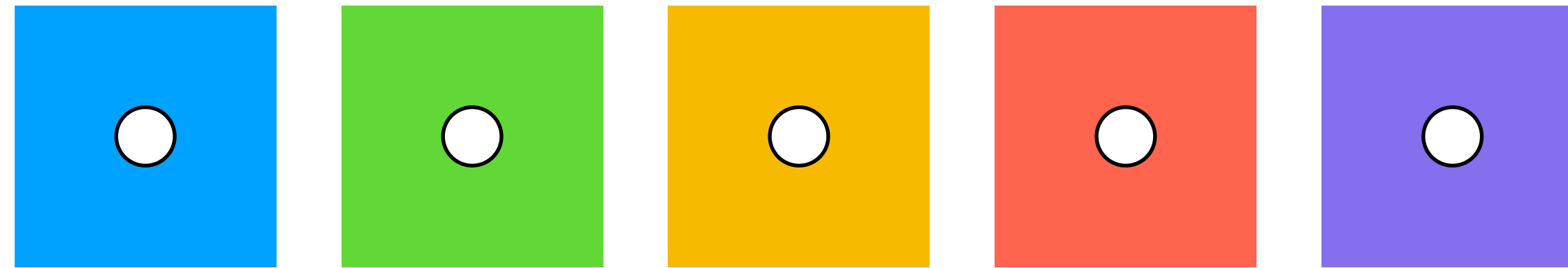
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



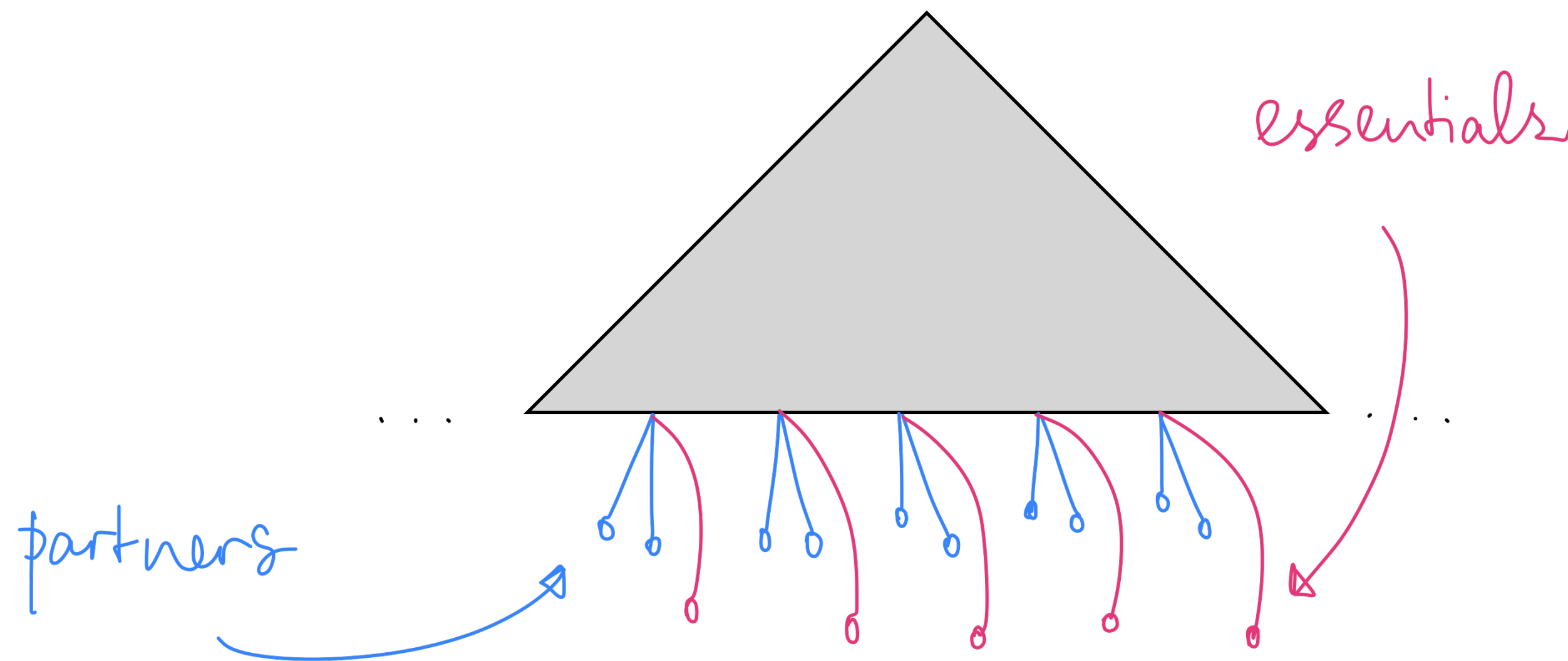
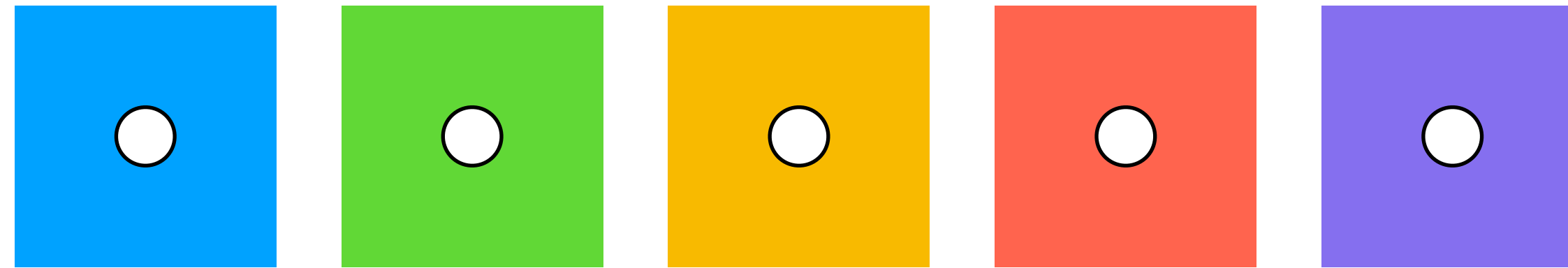
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



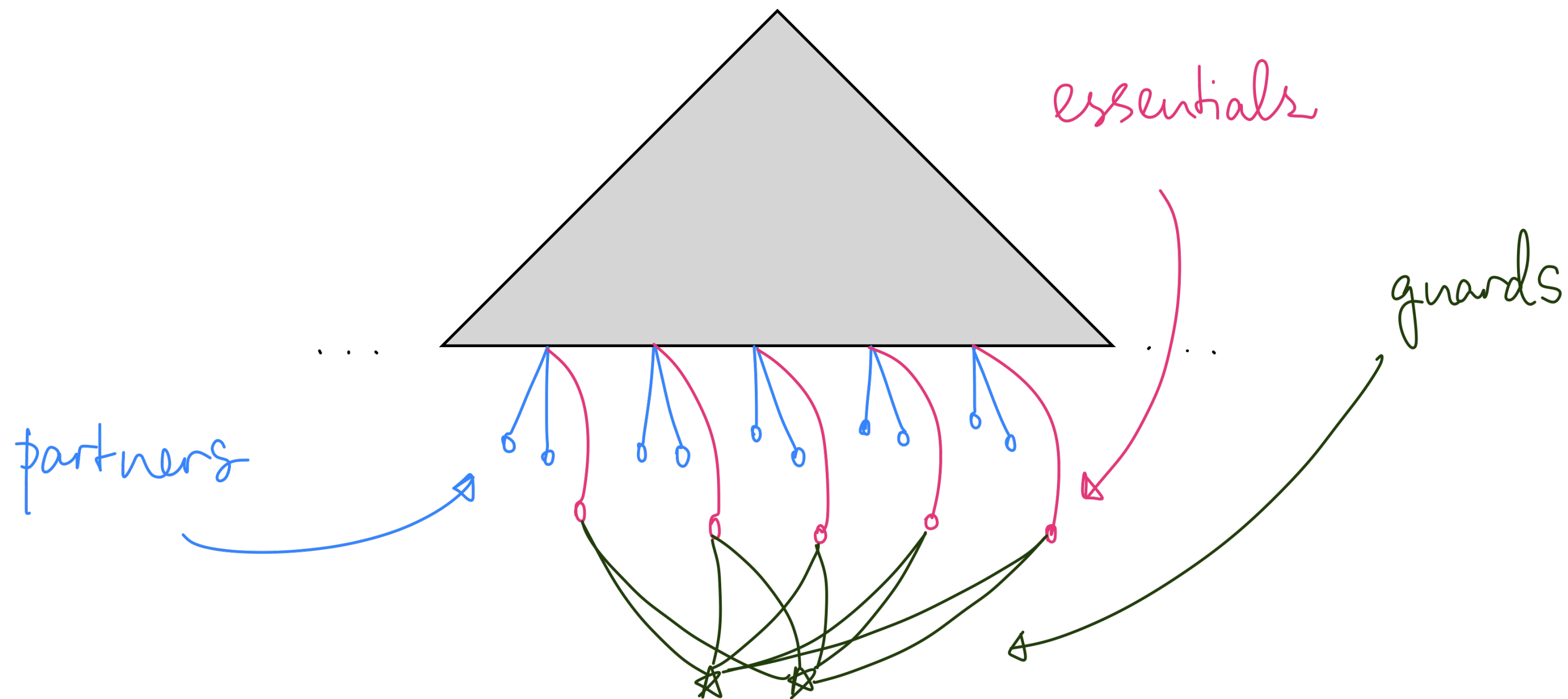
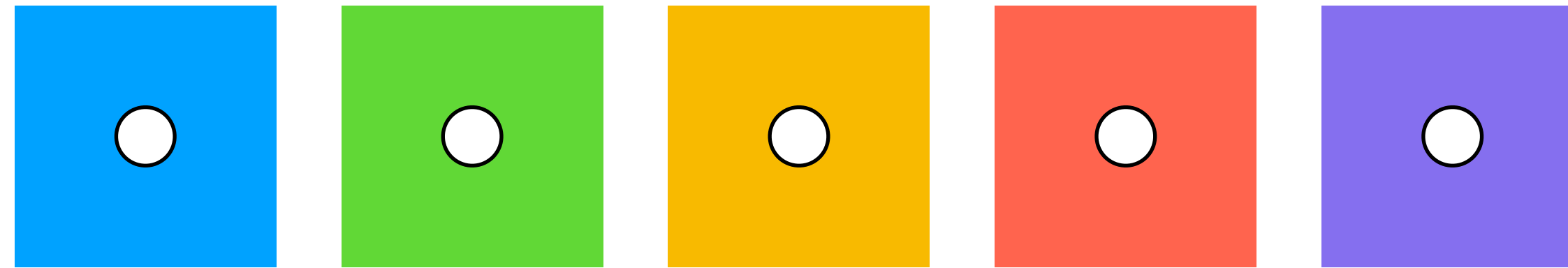
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



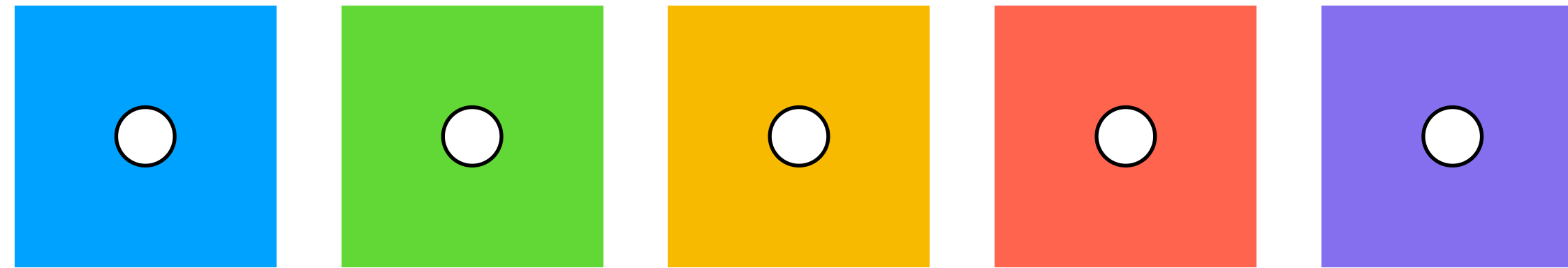
Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET

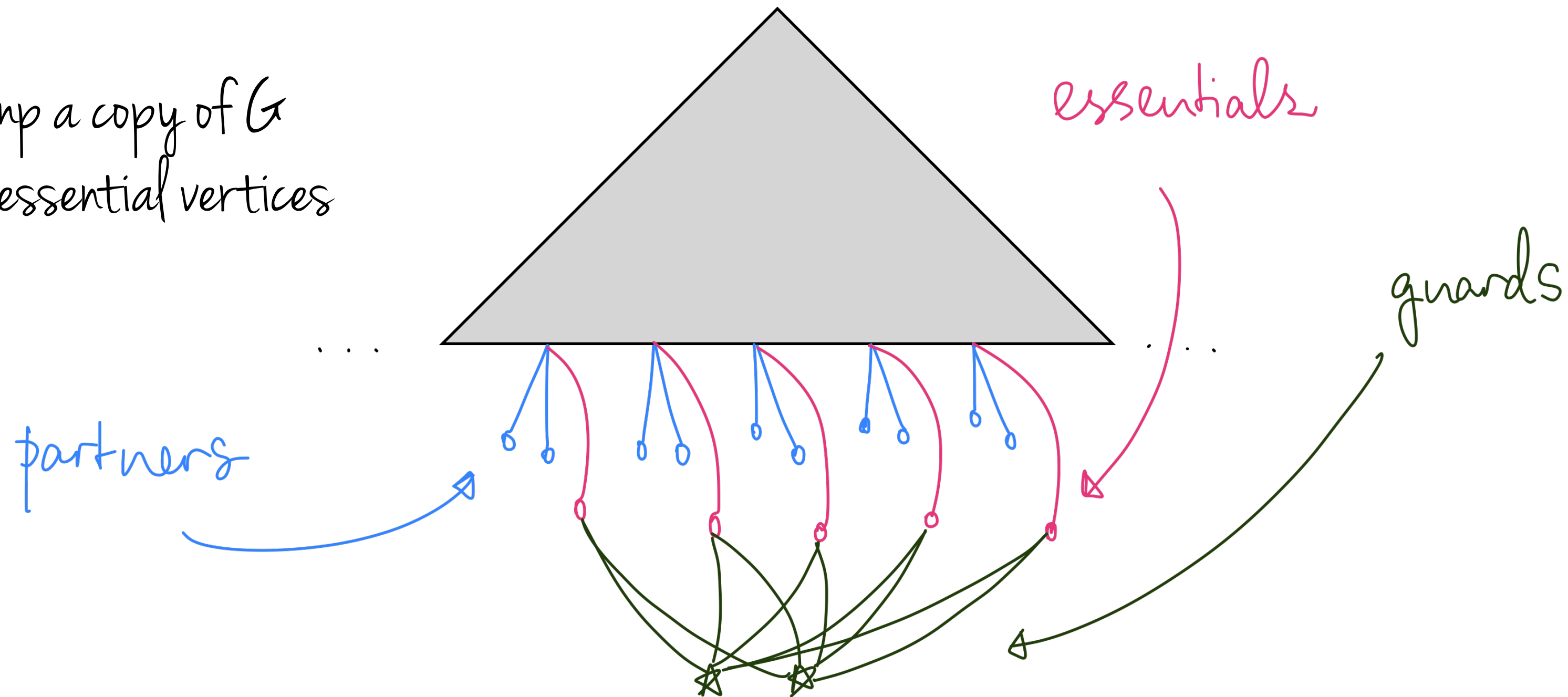


Vertex deletion to binary trees

Reduce from MULTI-COLORED INDEPENDENT SET



Dump a copy of G
on the essential vertices



Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

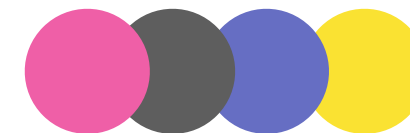
Vertex deletion to binary trees

Reduce from LINEAR NEAR-EXACT (2/2/4)-SAT



Core Clauses

length three; occur in pairs consisting of a main variable and two shadow variables
the shadow variables occur exactly once with negative polarity



Auxiliary clauses

length four; every shadow variable occurs exactly once with positive polarity

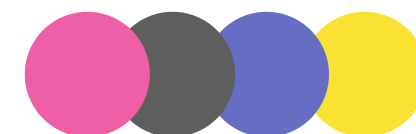
Vertex deletion to binary trees

Reduce from LINEAR NEAR-EXACT (2/2/4)-SAT



$$\mathcal{C} = \{A_1, B_1, A'_1, B'_1, \dots, A_p, B_p, A'_p, B'_p\} \cup \{C_1, \dots, C_q\}$$

$$\forall i \in [p], A_i \cap B_i = \{x_i\} \text{ and } A'_i \cap B'_i = \{\bar{x}_i\}$$



Auxiliary clauses

length four; every shadow variable occurs exactly once with positive polarity

Vertex deletion to binary trees

Reduce from LINEAR NEAR-EXACT (2/2/4)-SAT

Given a set of clauses as described before,
is there an assignment τ of truth values to the variables such that:

exactly one literal in every core clause
and two literals in every auxiliary clause

evaluate to true under τ ?

DETAILS OMITTED.

Overview

Introduction & Summary of Results

Hardness of deleting to ternary trees: a warm-up reduction

Deletion to binary trees - the vertex version

Deletion to binary trees - the edge version

Algorithmic questions & future directions

Algorithmic Questions and Future Directions

Algorithmic Questions and Future Directions

These problems are fixed-parameter tractable with respect to the standard parameter (solution size), by adapting ideas that have worked for FVS.

Algorithmic Questions and Future Directions

These problems are fixed-parameter tractable with respect to the standard parameter (solution size), by adapting ideas that have worked for FVS.

Informal Outline

Algorithmic Questions and Future Directions

These problems are fixed-parameter tractable with respect to the standard parameter (solution size), by adapting ideas that have worked for FVS.

Informal Outline

Branch on high-degree vertices and delete long degree-two paths.

Algorithmic Questions and Future Directions

These problems are fixed-parameter tractable with respect to the standard parameter (solution size), by adapting ideas that have worked for FVS.

Informal Outline

Branch on high-degree vertices and delete long degree-two paths.

Branch on short cycles (of length at most $\log(n)$).

Algorithmic Questions and Future Directions

Algorithmic Questions and Future Directions

Single-exponential FPT algorithms, perhaps by appropriately adapting the iterative compression approaches that have worked well for FVS and related problems.

Algorithmic Questions and Future Directions

Single-exponential FPT algorithms, perhaps by appropriately adapting the iterative compression approaches that have worked well for FVS and related problems.

Structural Parameters.

Algorithmic Questions and Future Directions

Single-exponential FPT algorithms, perhaps by appropriately adapting the iterative compression approaches that have worked well for FVS and related problems.

Structural Parameters.

Kernelization.

Algorithmic Questions and Future Directions

Single-exponential FPT algorithms, perhaps by appropriately adapting the iterative compression approaches that have worked well for FVS and related problems.

Structural Parameters.

Kernelization.

Approximation algorithms.

Thank You!

neeldhara.m@iitgn.ac.in

