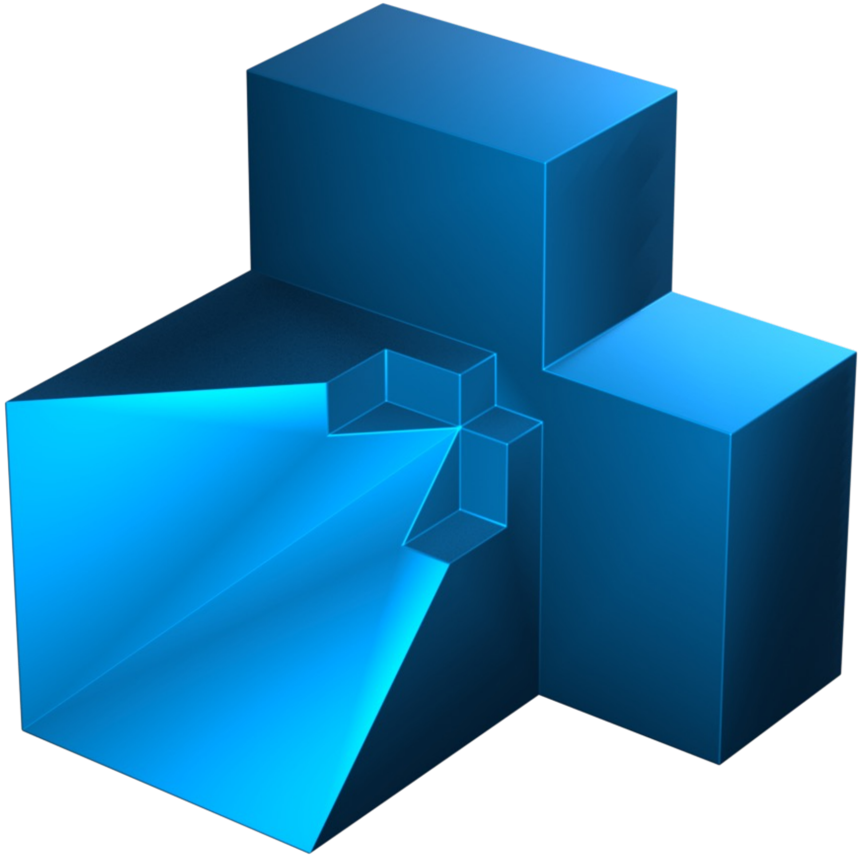
**Jukka Suomela**

Aalto University

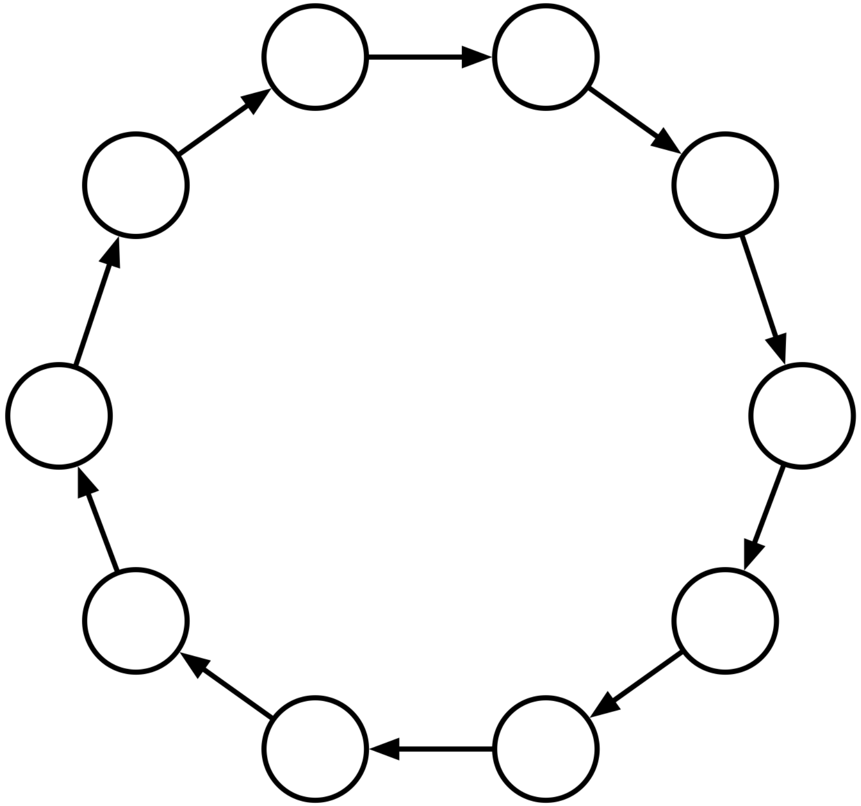
*joint work with many people*
*and many chatbots*

# 2-coloring in one round

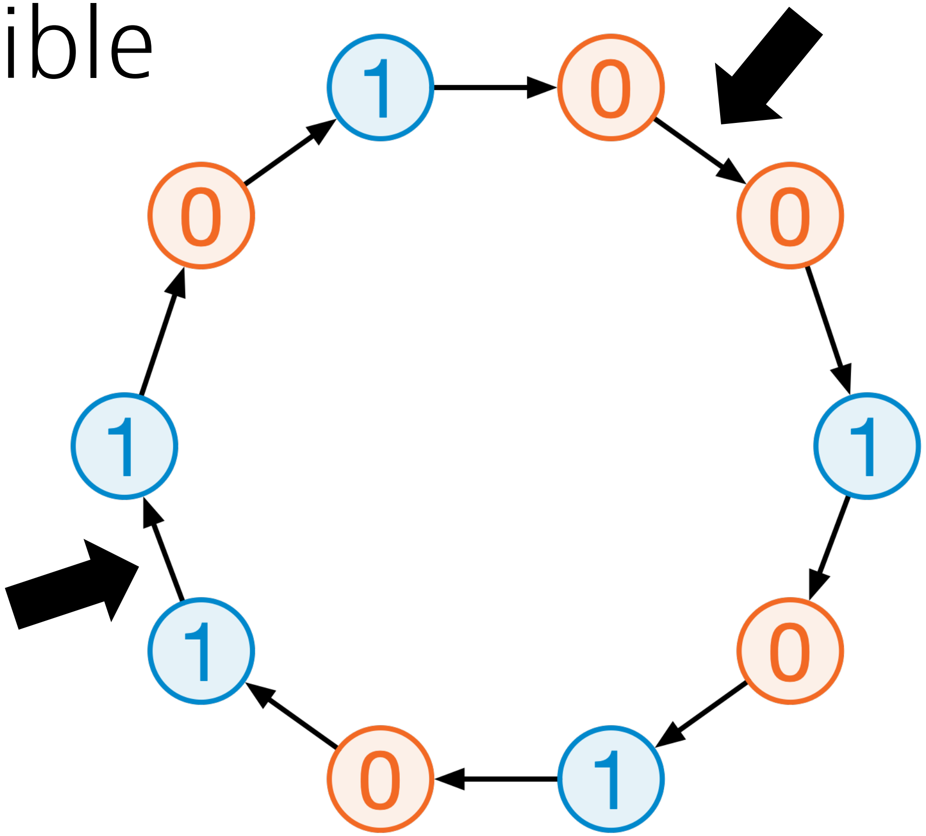
# Problem setting

- **Setting:** directed cycle, $n$ computers



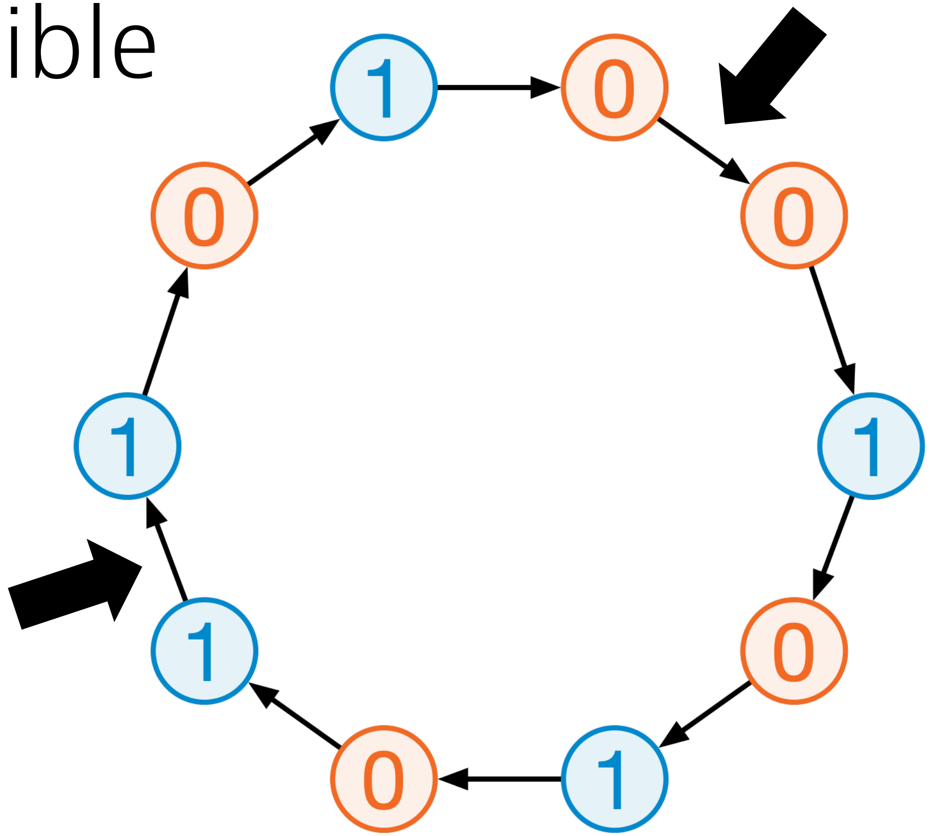
# Problem setting

- **Setting:** directed cycle, *n* computers
- **Task:** 2-color as well as possible
  - minimize the number of monochromatic edges



# Problem setting

- **Setting:** directed cycle, *n* computers
- **Task:** 2-color as well as possible
  - minimize the number of monochromatic edges
- **Model:** one-round distributed algorithms

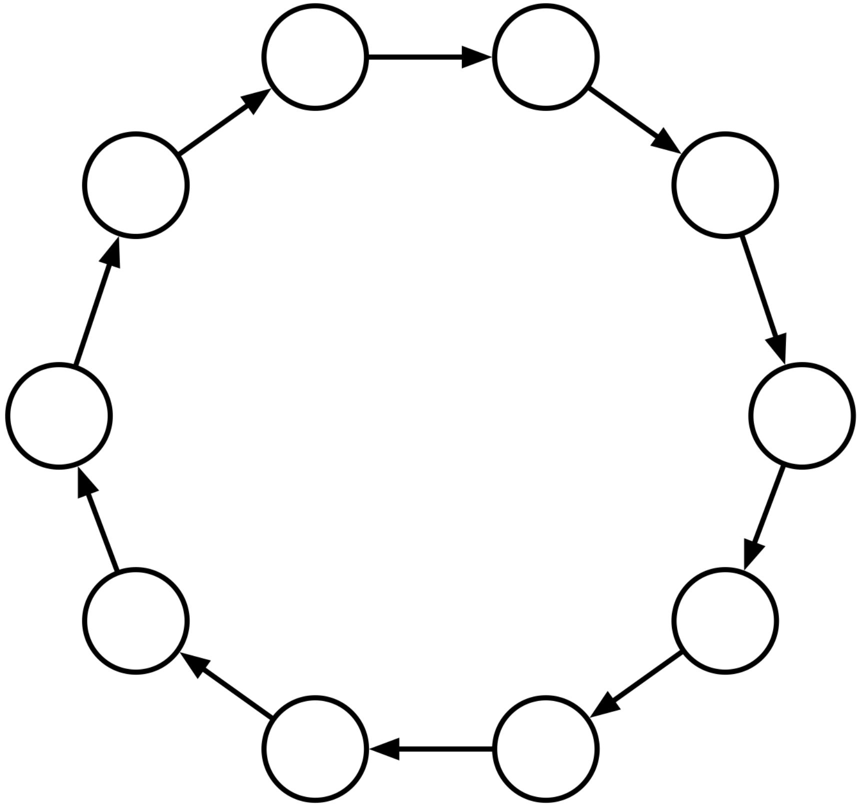


# Why?

- **Major open question:** understanding ***distributed quantum advantage***
  - symmetry breaking problems?
  - 3-coloring cycles?
  - constant-round quantum algorithms?
- **Toy question:** *1-round quantum algorithms for 2-coloring cycles?*

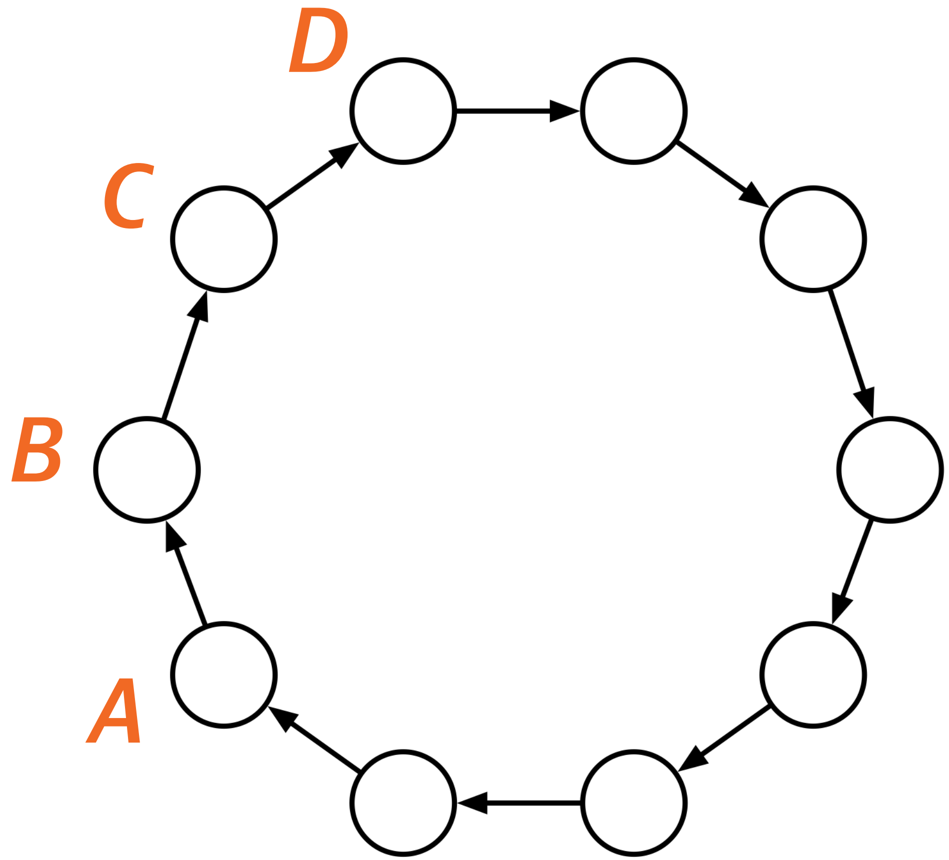
# What is the classical baseline??

# Precise setting



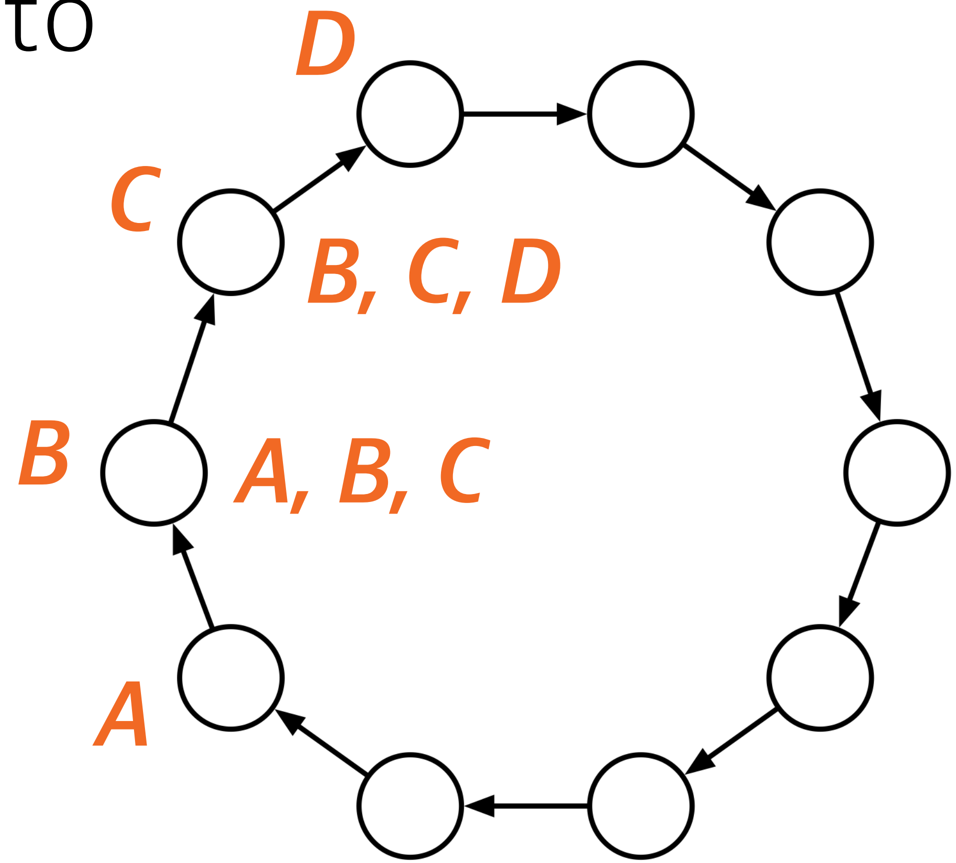
# Precise setting

- Each node produces independently a ***random number***



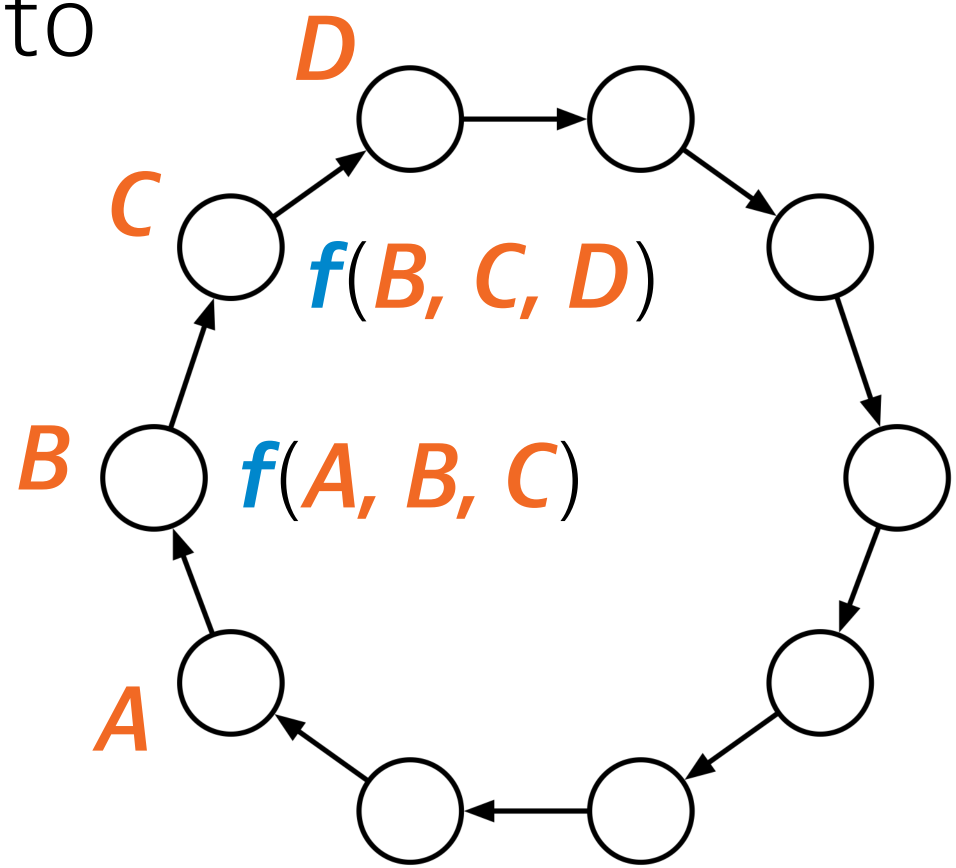
# Precise setting

- Each node produces independently a ***random number***, sends it to predecessor and successor
- You have 3 values ***A, B, C***



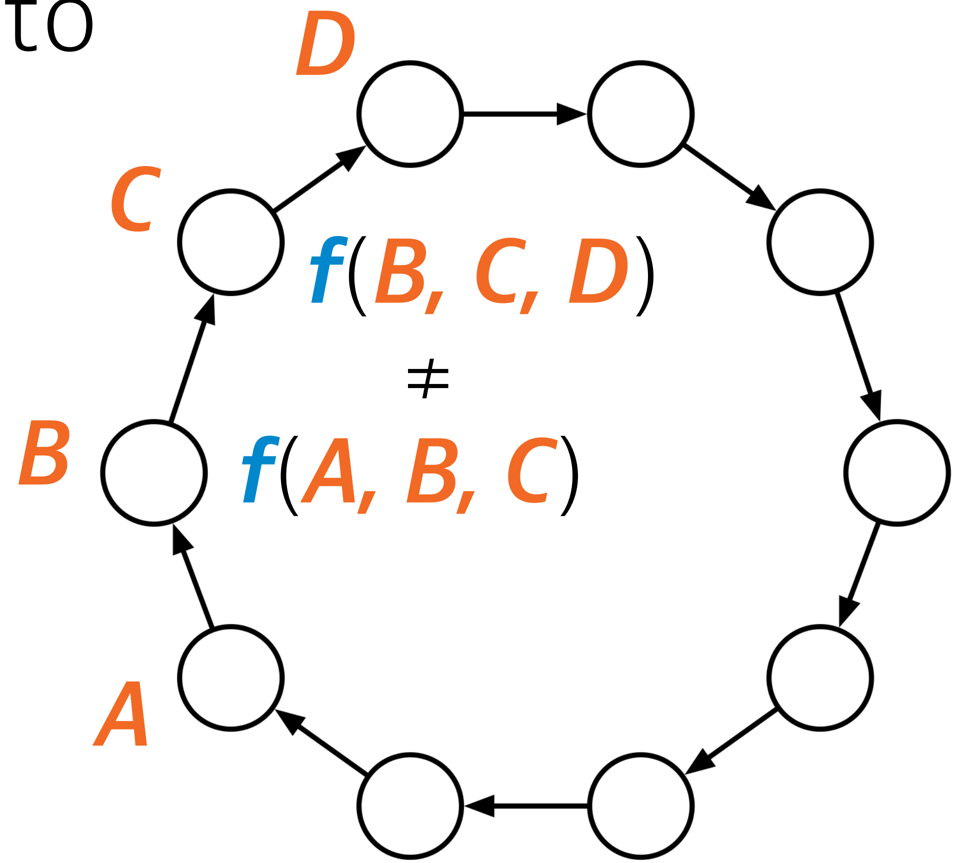
# Precise setting

- Each node produces independently a ***random number***, sends it to predecessor and successor
- You have 3 values ***A, B, C***
- Compute ***f(A, B, C)***



# Precise setting

- Each node produces independently a ***random number***, sends it to predecessor and successor
- You have 3 values *A, B, C*
- Compute $f(A, B, C)$
- Ideally forms a proper 2-coloring



# Formally

$$f\colon [0,1]^3 \to \{0,1\}$$

$$A, B, C, D \sim \text{Uniform}\ [0,1]$$

$$\boldsymbol{p = \Pr[f(A, B, C) = f(B, C, D)]}$$

*How small can you make probability p by choosing the best possible function f?*

# Trivial: $p \leq 1/2$

- $f(A, B, C) = 1$ iff $B \geq 1/2$
- Independent coin flips, random 2-coloring
- Monochromatic with probability 1/2

# Simple: $p \le 1/3$

- **Idea:** produce large independent sets
  - no 1–1 conflicts, only 0–0 conflicts
- **Algorithm:** local maxima join:
  $f(A, B, C) = 1$ iff $B > A$ and $B > C$
- **Analysis:** $A, B, C, D$ bad iff
  $B$ and $C$ not local maxima:
  $p = 2/4 \cdot 2/3 = 1/3$

**1** = largest
**2** = $2^{nd}$ largest

| | | | |
|---|---|---|---|
| 1 | 2 | | |
| 1 | | | 2 |
| 2 | | | 1 |
| | | 2 | 1 |

# Simple: $p \geq 1/5$

- Apply your function $f$ in a 5-cycle
- There is always at least 1 monochromatic edge

$$1/5 \leq p \leq 1/3$$

# Systematic approaches?

# De Bruijn graphs

*a*, *b*, *c*, *d*:
{0, 1, …, *n* − 1}

## "Normal"

- node (*a*, *b*, *c*)
  for any *a*, *b*, *c*
- edge (*a*, *b*, *c*) → (*b*, *c*, *d*)
  for any *a*, *b*, *c*, *d*

**0, 1, 1** → **1, 1, 2**

## "Distinct"

- node (*a*, *b*, *c*)
  for any ***distinct*** *a*, *b*, *c*
- edge (*a*, *b*, *c*) → (*b*, *c*, *d*)
  for any ***distinct*** *a*, *b*, *c*, *d*

**0, 1, 3** → **1, 3, 2**

# Cuts in De Bruijn graphs

- How well can you 2-color De Bruijn graphs?
- What is the smallest fraction of monochromatic edges?
- $p_{\mathrm{normal}}(n)$ = optimum in "normal" DB graphs
- $p_{\mathrm{distinct}}(n)$ = optimum in "distinct" DB graphs

# Cuts in De Bruijn graphs

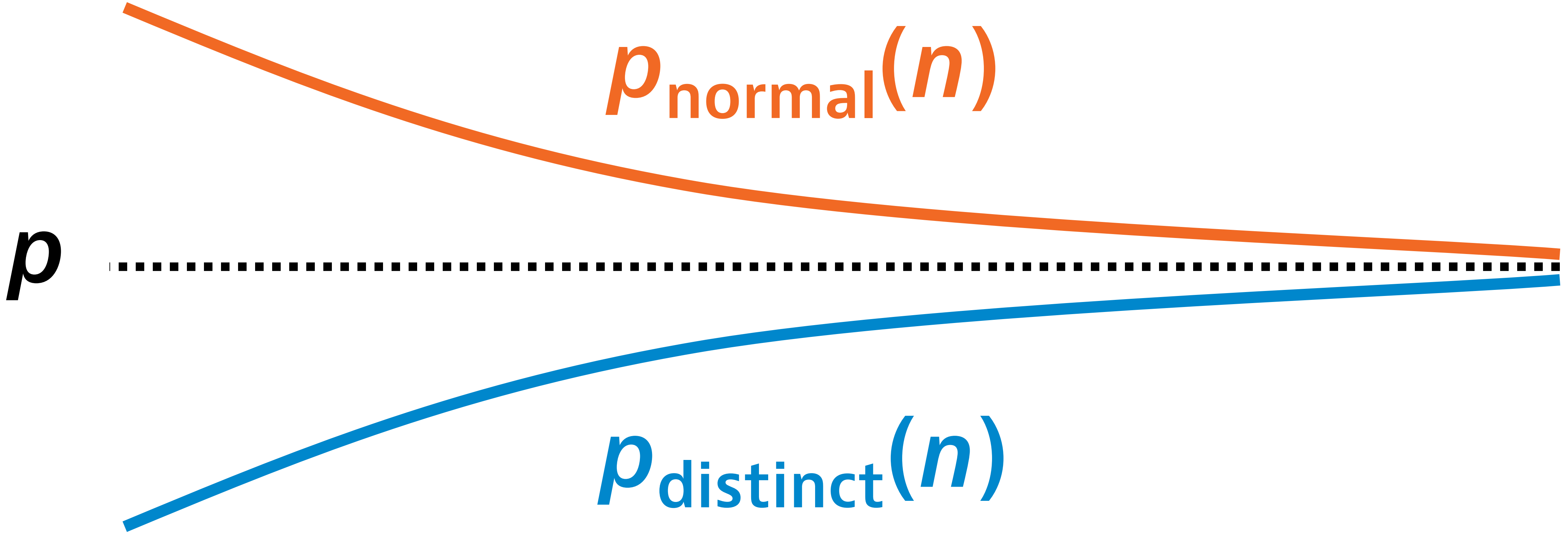
$$p_{\text{distinct}}(n) \leq p \leq p_{\text{normal}}(n)$$

**Lower bounds**

**Algorithms**

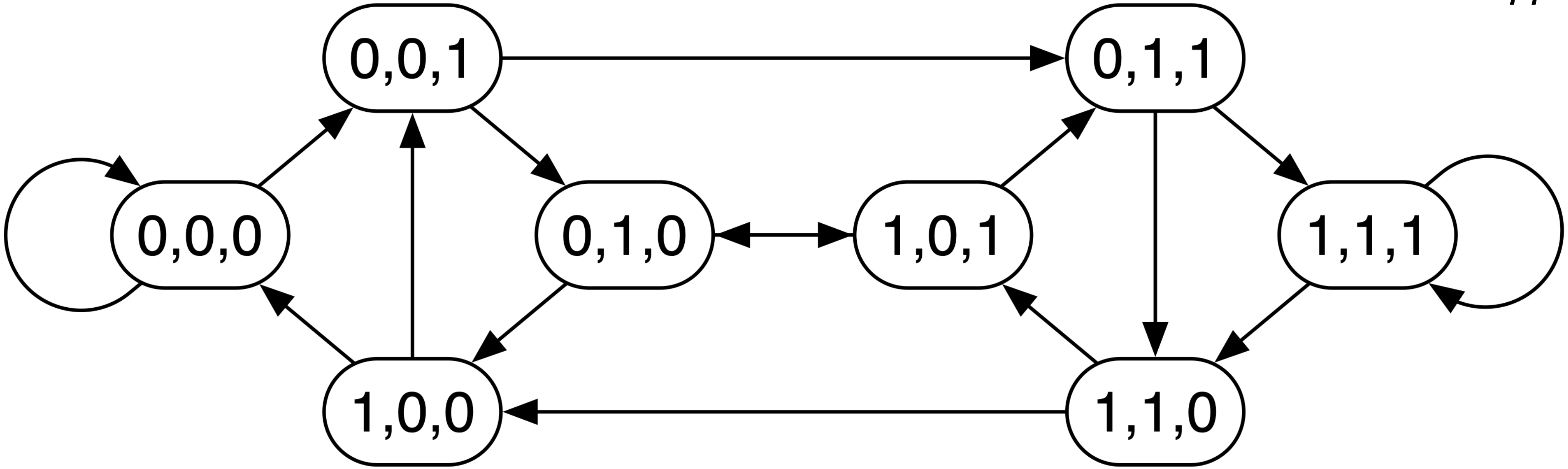
$$p_{\text{distinct}}(n) \to p_{\text{normal}}(n)$$

# Cuts in De Bruijn graphs

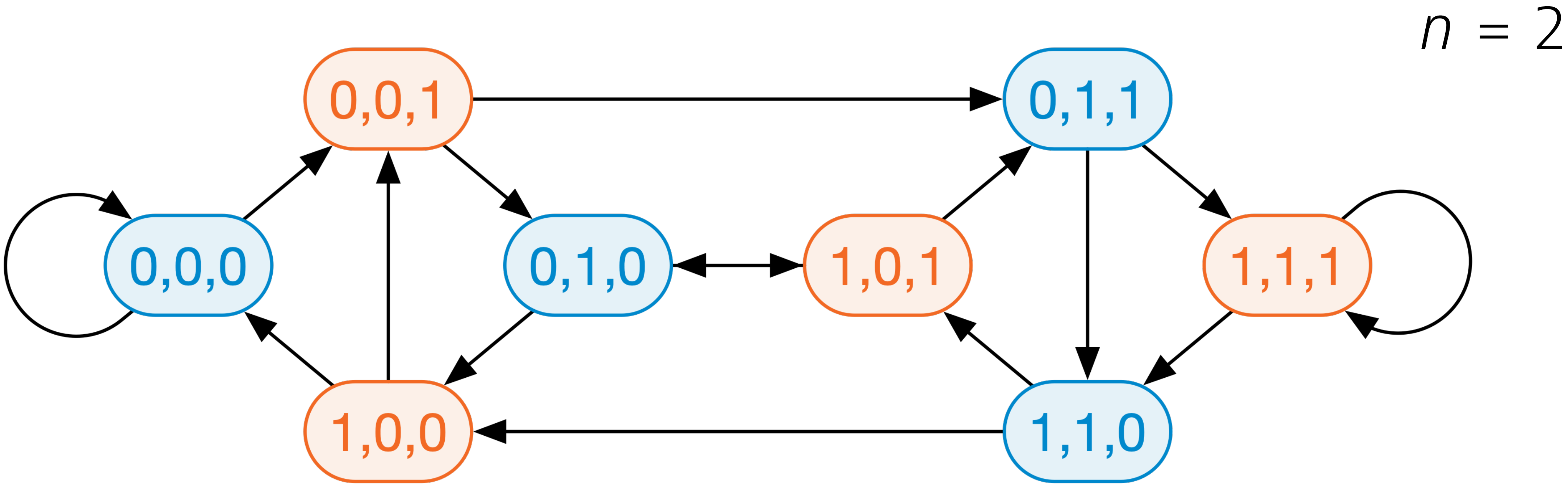


# De Bruijn → algorithm

$n = 2$



# De Bruijn → algorithm




$p_{\text{normal}}(2) = 1/4$ → $p \leq 1/4$

# De Bruijn → lower bound

- Pick $A, B, C, D$
- See if $f(A, B, C) = f(B, C, D)$

# De Bruijn → lower bound

- Pick $X_0, \ldots, X_{n-1}$
- Pick distinct indexes $A, B, C, D$
- See if $f(X_A, X_B, X_C) = f(X_B, X_C, X_D)$

# De Bruijn → lower bound

- Pick $X_0, \ldots, X_{n-1}$
- Node $(a, b, c)$ has color $f(X_a, X_b, X_c)$
- Pick distinct indexes $A, B, C, D$
- See if $f(X_A, X_B, X_C) = f(X_B, X_C, X_D)$

Distinct De Bruijn graph

# De Bruijn → lower bound

- Pick $X_0, \ldots, X_{n-1}$
- Node $(a, b, c)$ has color $f(X_a, X_b, X_c)$
- Pick random edge $(A, B, C) \to (B, C, D)$
- See if $f(X_A, X_B, X_C) = f(X_B, X_C, X_D)$

**Distinct De Bruijn graph**

# De Bruijn → lower bound

- Pick $X_0, \ldots, X_{n-1}$
- Node $(a, b, c)$ has color $f(X_a, X_b, X_c)$
- Pick random edge $e$
- See if $e$ is monochromatic

Distinct De Bruijn graph

# De Bruijn → lower bound

- Pick $X_0, \ldots, X_{n-1}$
- Node $(a, b, c)$ has color $f(X_a, X_b, X_c)$
- Pick random edge $e$
- See if $e$ is monochromatic
- Recall: every 2-coloring has fraction $\geq p_{\text{distinct}}(n)$ monochromatic edges

Distinct De Bruijn graph

# De Bruijn → lower bound

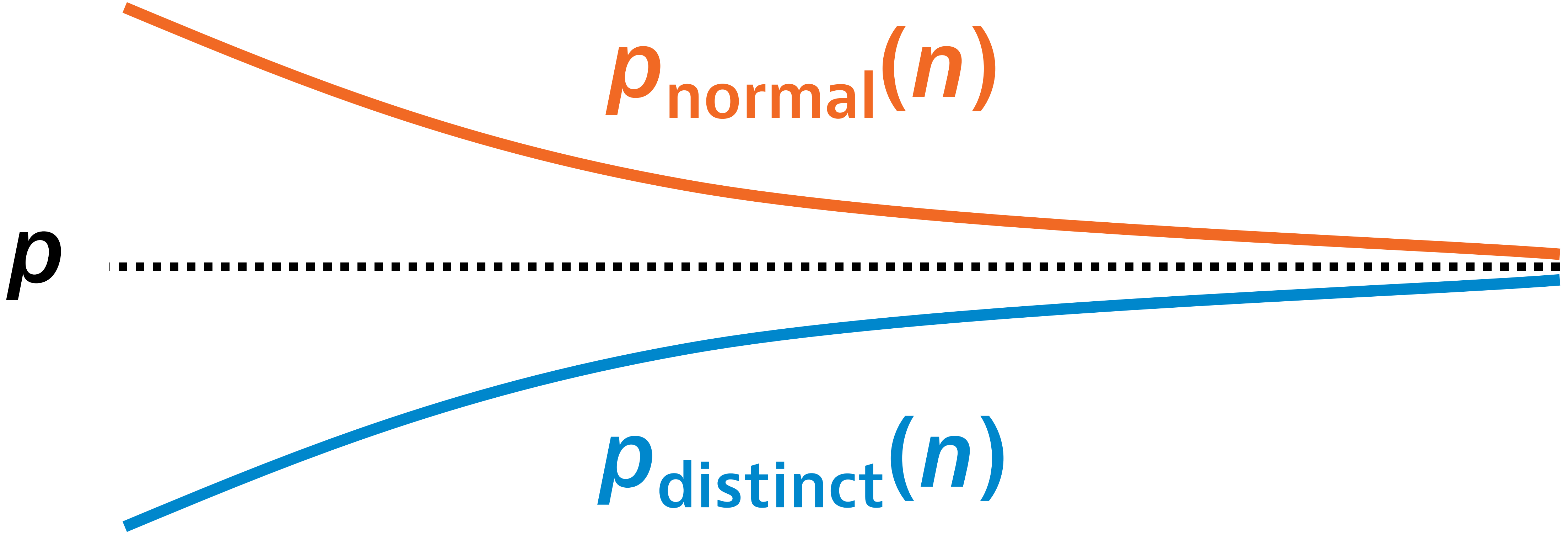
- Pick $X_0, \ldots, X_{n-1}$
- Node $(a, b, c)$ has color $f(X_a, X_b, X_c)$
- Pick random edge $e$
- See if $e$ is monochromatic
- $p \geq p_{\text{distinct}}(n)$

Distinct De Bruijn graph

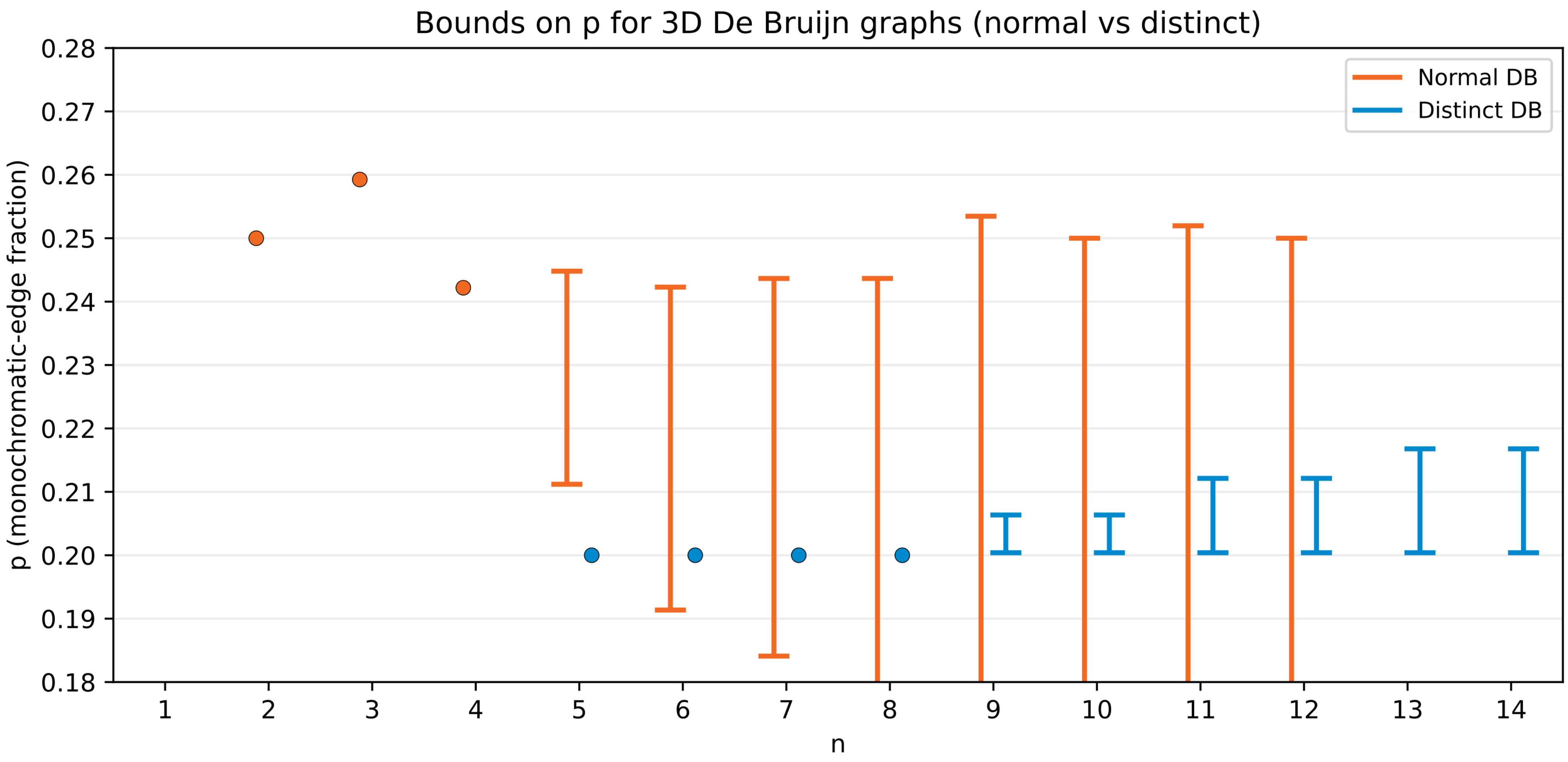
$p_{\text{distinct}}(6) = 1/5$

→ $p \geq 1/5$

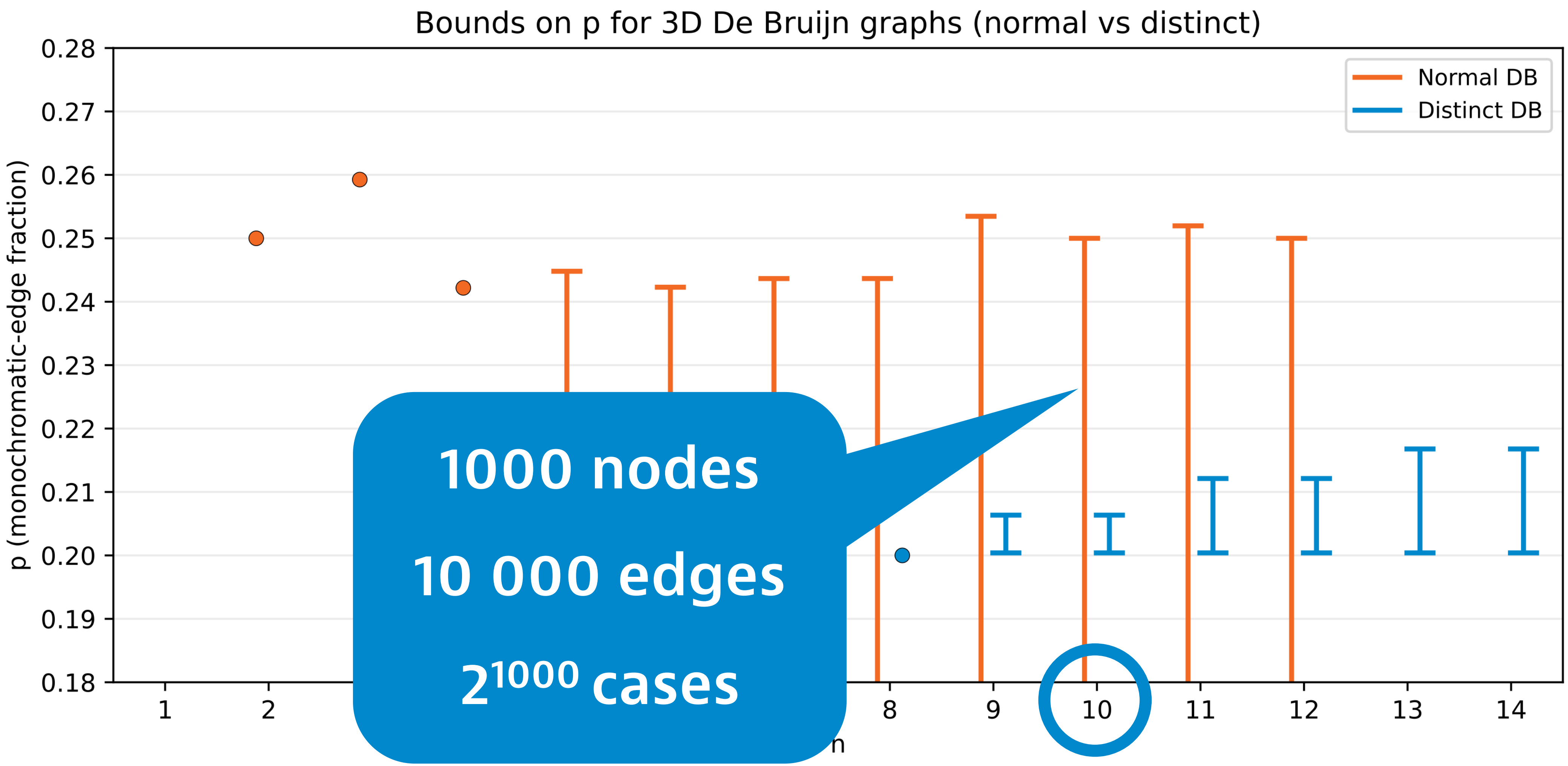
# Cuts in De Bruijn graphs



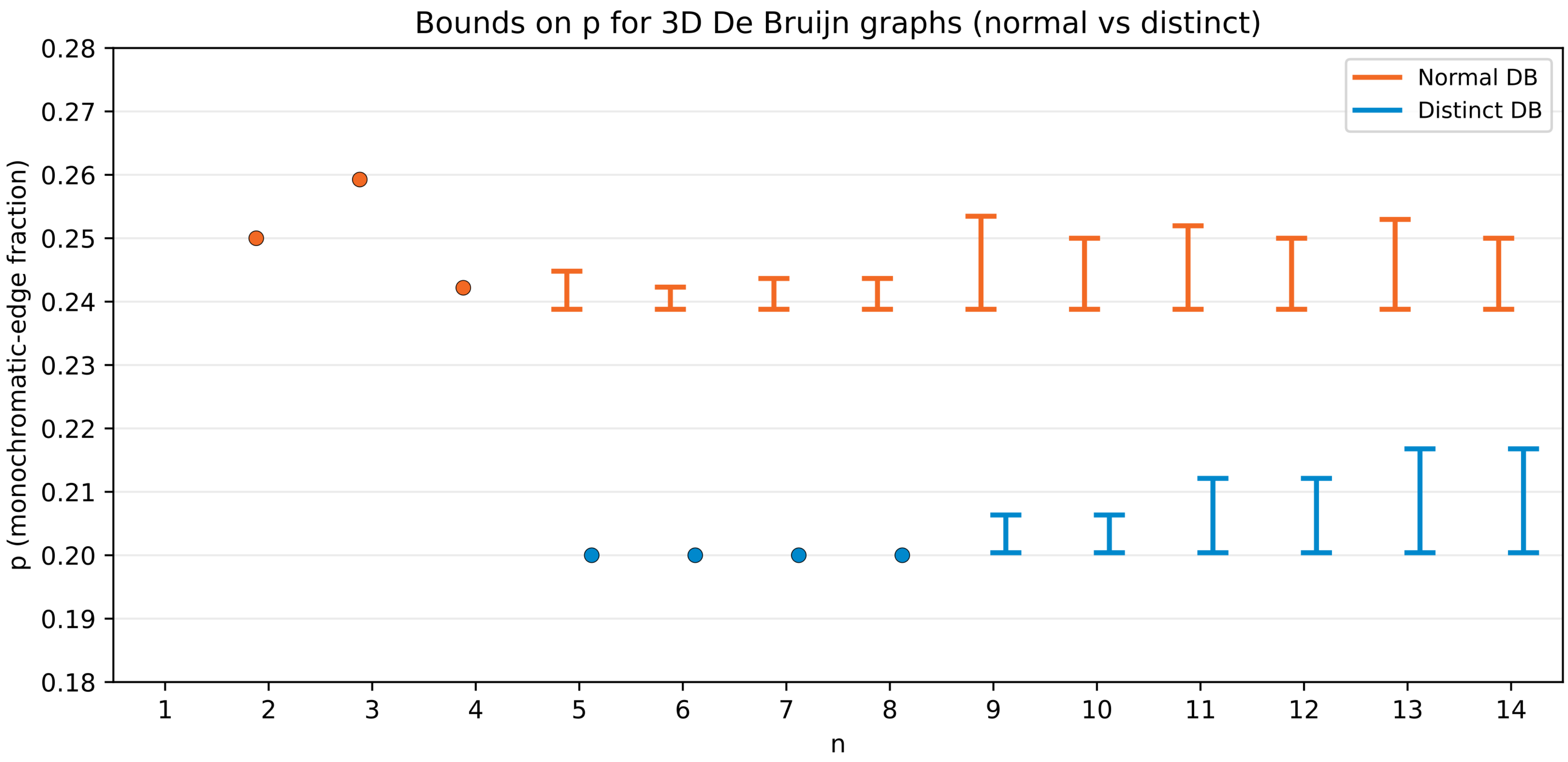
# Cuts in De Bruijn graphs



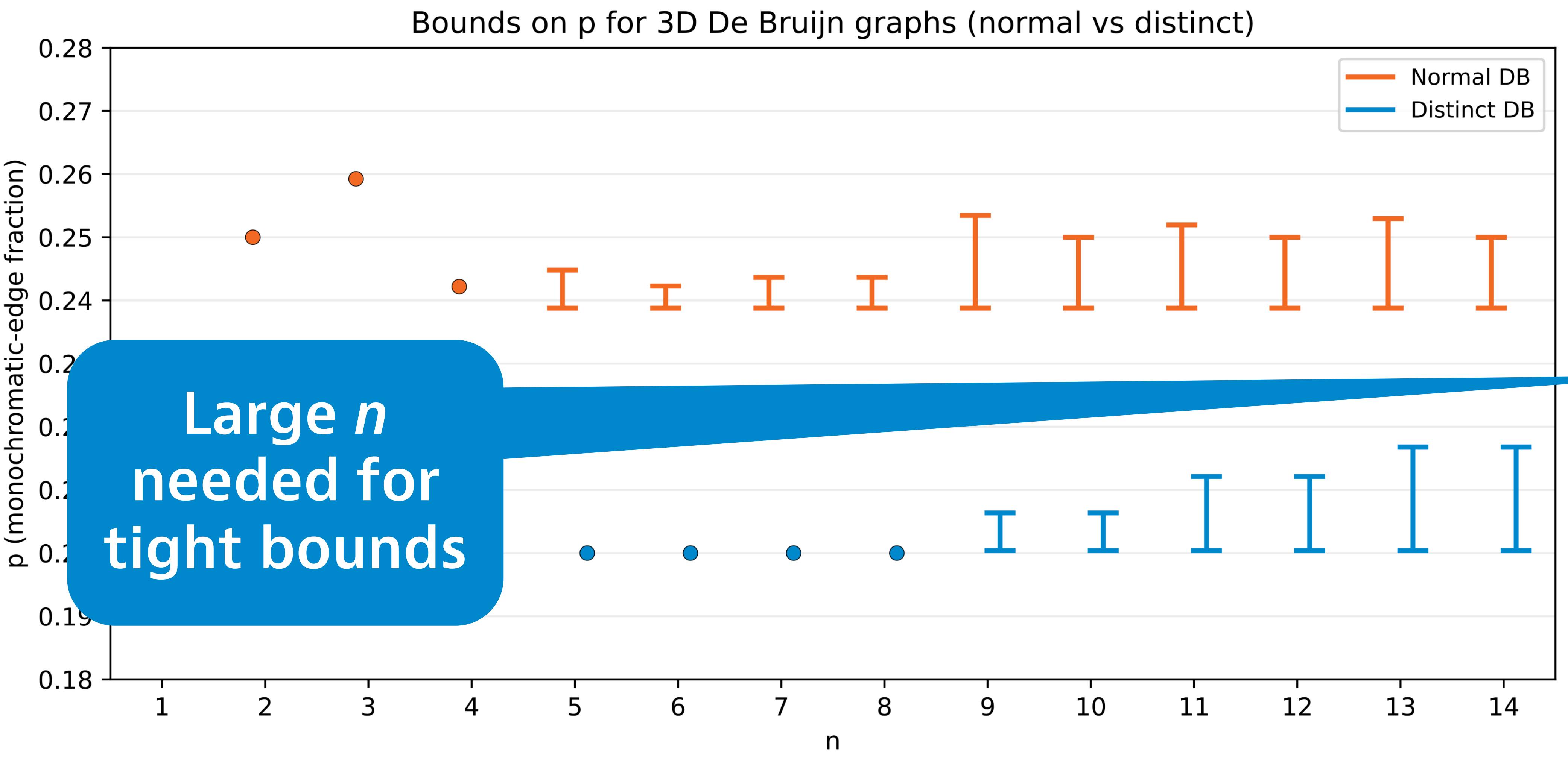
# Cuts in De Bruijn graphs



# Cuts in De Bruijn graphs



# Cuts in De Bruijn graphs




Large *n* needed for tight bounds

# Chatbots can help

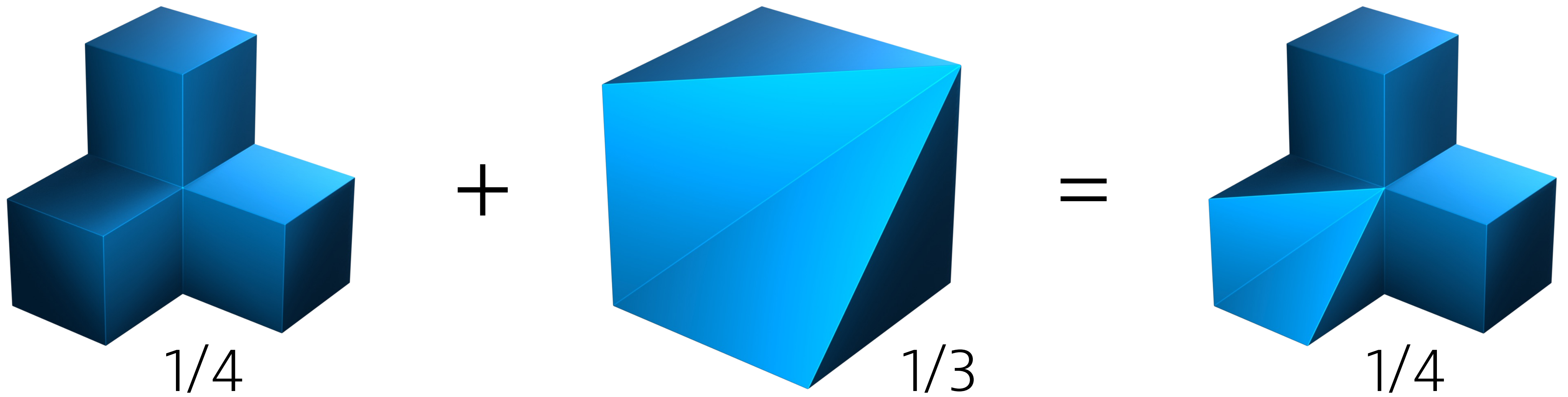
- Good estimate for $p_{\text{distinct}}(n)$ for $n$ **= 1 million**
- GPT-5.2 solved it like this:
  - ***SDP relaxation*** + "correlation triangle inequalities"
  - ***Symmetries:*** 34 overlap types, 479 triangle types
  - Solve SDP, find a feasible ***dual*** solution that lower-bounds primal optimum
  - Argue that this also lower-bounds best possible monochromatic fraction over all 2-colorings …

# Chatbots can help

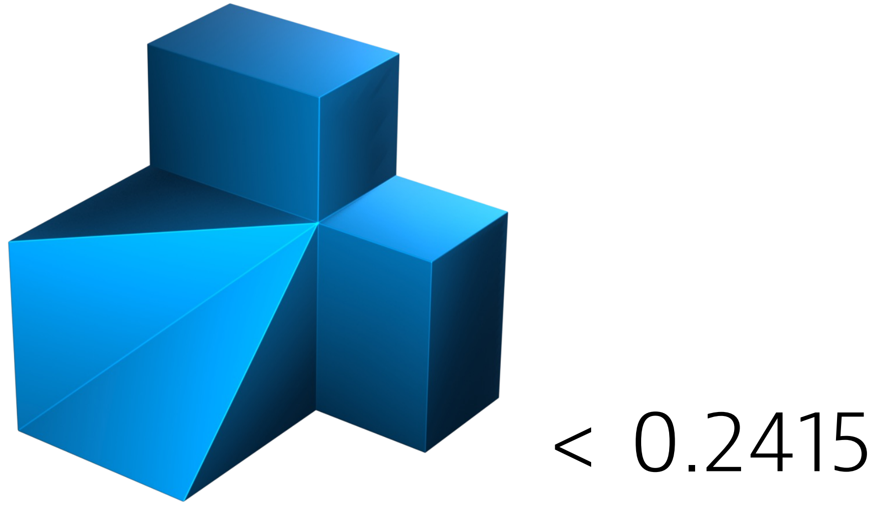
- Good estimate for $p_{\text{distinct}}(n)$ for $n$ **= 1 million**
- GPT-5.2 solved it like this:
  - [*too complicated for me to verify*]
- GPT-5.2 also **formalized the proof in Lean 4**
  - 12 000+ lines of Lean 4 code

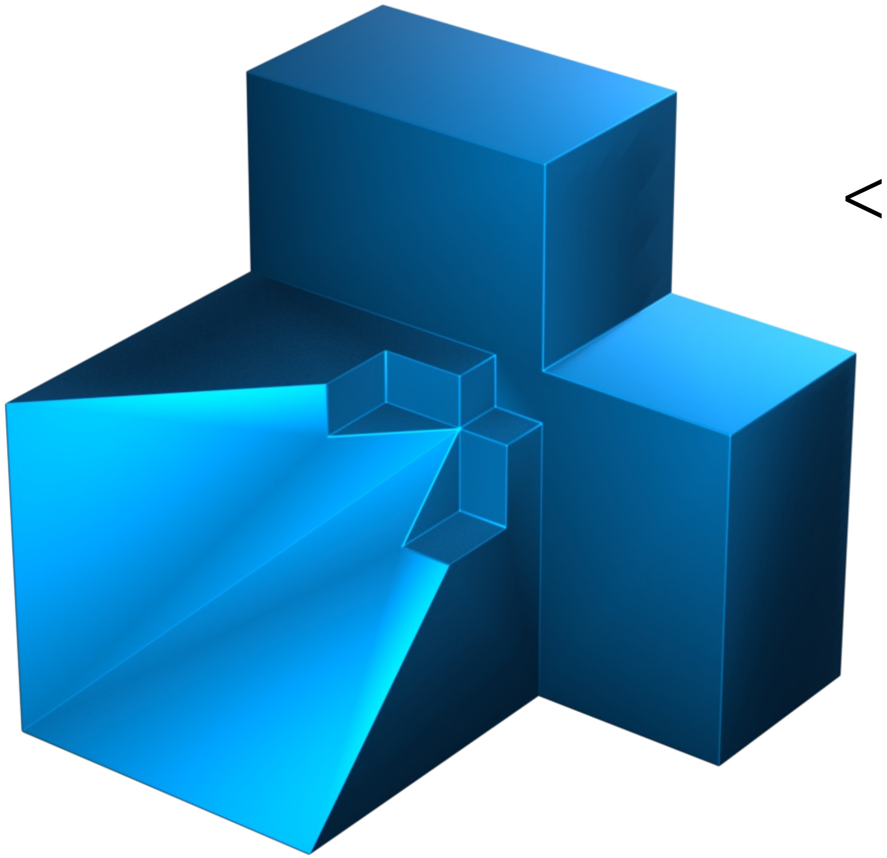
# Chatbots can help

- Good estimate for $p_{\text{distinct}}(n)$ for $n$ **= 1 million**
- GPT-5.2 solved it like this:
  - [*too complicated for me to verify*]
- GPT-5.2 also **formalized the proof in Lean 4**
  - 12 000+ lines of Lean 4 code
- $p \geq p_{\text{distinct}}(1\,000\,000) \geq \underline{\underline{0.23879}}$



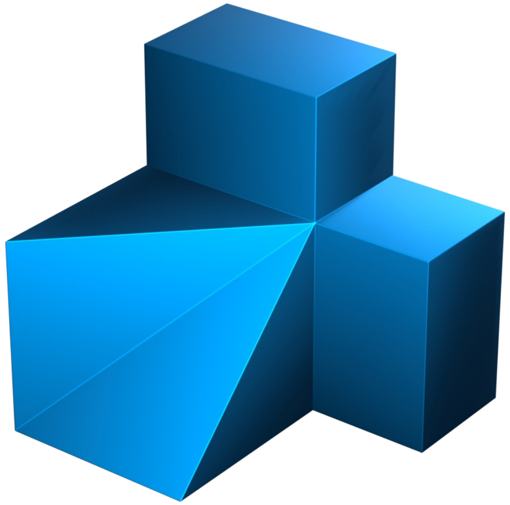

# Upper bounds





< 0.24118

# Upper bounds



< 0.2415

# 1-round 2-coloring

- **0.23879 ≤ *p* < 0.24118**

Trivial: $0.2 \le p \le 0.25$

- Most of the work done by **Codex + GPT-5.2**
- Upper & lower bounds formalized in **Lean 4**
  - 17 000+ lines of Lean code total

```
theorem pStar_ge_23879 :
  ENNReal.ofReal (23879 / 100000 : ℝ) ≤ ClassicalAlgorithm.pStar := ...

theorem pStar_lt_24118 :
  ClassicalAlgorithm.pStar < ENNReal.ofReal (24118 / 100000 : ℝ) := ...
```