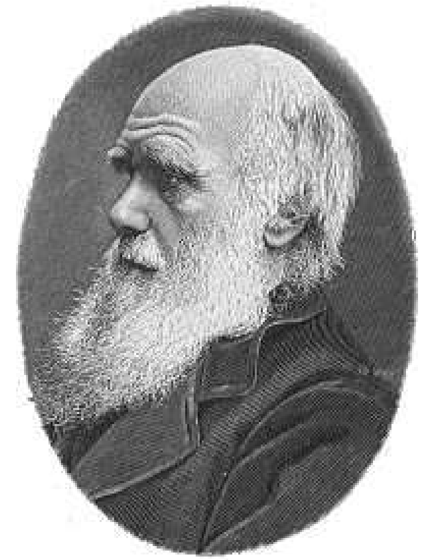
# Binary Contingency Tables in Theory and Practice

Ivona Bezáková

(Rochester Institute of Technology)

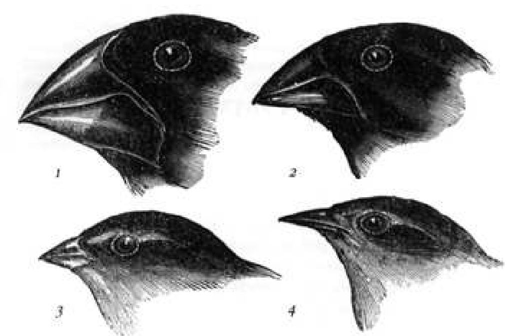
Based on joint works with Nayantara Bhatnagar, Alistair Sinclair, Daniel Štefankovič, and Eric Vigoda.

# Darwin's Finches



The Voyage of the Beagle

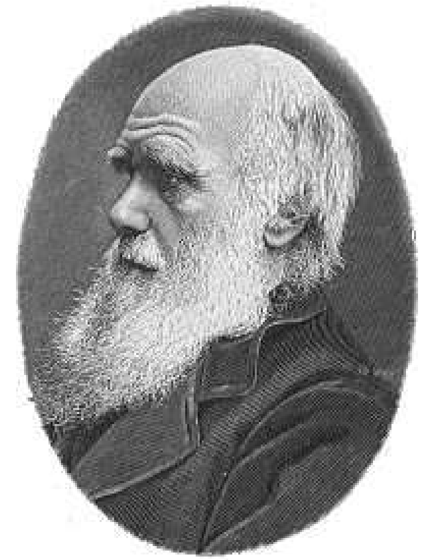
Galápagos archipelago (1835)

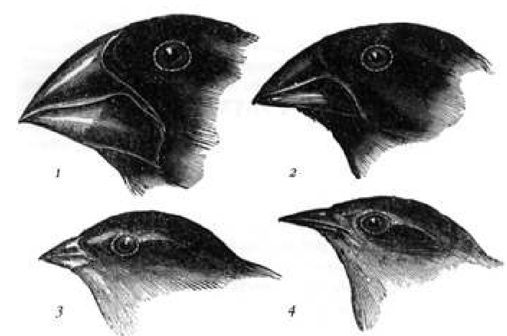


# Darwin's Finches

**Distribution of Darwin's Finches on Visitor Islands**

| | Santa Cruz | Plaza | Santa Fe | San Cristobal | Espanola | Floreana | Isabela | Fernandina | Santiago | Rabida |
|---|---|---|---|---|---|---|---|---|---|---|
| Small Ground Finch | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Medium Ground Finch | ● | ● | ● | ● | | ● | ● | ● | ● | ● |
| Large Ground Finch | ● | | | | | | ● | ● | ● | ● |
| Cactus Ground Finch | ● | ● | ● | ● | | ● | ● | | ● | ● |
| Large Cactus Ground Finch | | | | | ● | | | | | |
| Sharp-beaked Ground Finch | | | | | | | | ● | ● | |
| Vegetarian Finch | ● | | | ● | | ● | ● | ● | ● | ● |
| Small Tree Finch | ● | | ● | ● | | ● | ● | ● | ● | ● |
| Medium Tree Finch | | | | | | ● | | | | |
| Large Tree Finch | ● | | | | | ● | ● | ● | ● | ● |
| Woodpecker Finch | ● | | | ● | | | ● | | ● | |
| Mangrove Finch | | | | | | | ● | ● | | |
| Warbler Finch | ● | | ● | ● | ● | ● | ● | ● | ● | ● |





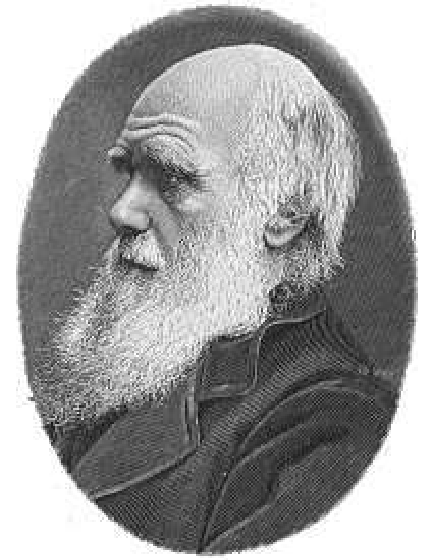
# Darwin's Finches

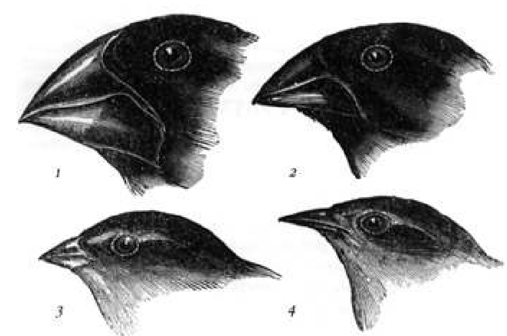
Distribution of Darwin's Finches on Visitor Islands

| | Santa Cruz | Plaza | Santa Fe | San Cristobal | Espanola | Floreana | Isabela | Fernandina | Santiago | Rabida |
|---|---|---|---|---|---|---|---|---|---|---|
| Small Ground Finch | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| Medium Ground Finch | ● | ● | ● | ● | | ● | ● | ● | ● | ● |
| Large Ground Finch | ● | | | | | | ● | ● | ● | ● |
| Cactus Ground Finch | ● | ● | ● | ● | | ● | ● | | ● | ● |
| Large Cactus Ground Finch | | | | | ● | | | | | |
| Sharp-beaked Ground Finch | | | | | | | | ● | ● | |
| Vegetarian Finch | ● | | | ● | | ● | ● | ● | ● | ● |
| Small Tree Finch | ● | | ● | ● | | ● | ● | ● | ● | ● |
| Medium Tree Finch | | | | | | ● | | | | |
| Large Tree Finch | ● | | | | | ● | ● | ● | ● | ● |
| Woodpecker Finch | ● | | | ● | | | ● | | ● | |
| Mangrove Finch | | | | | | | ● | ● | | |
| Warbler Finch | ● | | ● | ● | ● | ● | ● | ● | ● | ● |

8

10

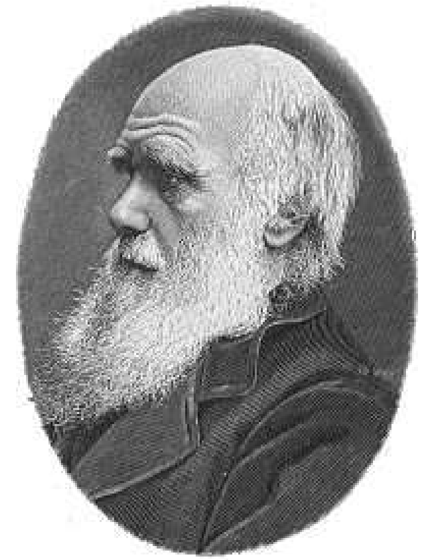




# Darwin's Finches

**Distribution of Darwin's Finches on Visitor Islands**

| | Santa Cruz | Plaza | Santa Fe | San Cristobal | Espanola | Floreana | Isabela | Fernandina | Santiago | Rabida | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Small Ground Finch | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | 10 |
| Medium Ground Finch | ● | ● | ● | ● | | ● | ● | ● | ● | ● | 9 |
| Large Ground Finch | ● | | | | | | ● | ● | ● | ● | 6 |
| Cactus Ground Finch | ● | ● | ● | ● | | ● | ● | | ● | ● | 8 |
| Large Cactus Ground Finch | | | | | ● | | | | | | 2 |
| Sharp-beaked Ground Finch | | | | | | | | ● | ● | | 3 |
| Vegetarian Finch | ● | | | ● | | ● | ● | ● | ● | ● | 7 |
| Small Tree Finch | ● | | ● | ● | | ● | ● | ● | ● | ● | 8 |
| Medium Tree Finch | | | | | | ● | | | | | 1 |
| Large Tree Finch | ● | | | | | ● | ● | ● | ● | ● | 6 |
| Woodpecker Finch | ● | | | ● | | | ● | | ● | | 4 |
| Mangrove Finch | | | | | | | ● | ● | | | 2 |
| Warbler Finch | ● | | ● | ● | ● | ● | ● | ● | ● | ● | 10 |
| | 9 | 3 | 5 | 7 | 3 | 8 | 10 | 9 | 10 | 8 | |



chance

OR

competitive pressures

?

# Binary Contingency Tables

Input: marginals
(row sums $r_1, r_2, \ldots, r_m$, column sums $c_1, c_2, \ldots, c_n$)

Sample space: 0/1 tables satisfying the marginals

Goal: count / sample

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | 4 |
| | | | | | | | 2 |
| | | | | | | | 3 |
| | | | | | | | 5 |
| | | | | | | | 3 |
| 3 | 4 | 2 | 1 | 2 | 2 | 3 | |

# Binary Contingency Tables

Input: marginals
(row sums $r_1, r_2, \ldots, r_m$, column sums $c_1, c_2, \ldots, c_n$)

Sample space: 0/1 tables satisfying the marginals

Goal: count / sample

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | | | 1 | | 1 | 4 |
| | 1 | 1 | | | | | 2 |
| 1 | 1 | | | | 1 | | 3 |
| | 1 | 1 | | 1 | 1 | 1 | 5 |
| 1 | | | 1 | | | 1 | 3 |
| 3 | 4 | 2 | 1 | 2 | 2 | 3 | |

# Different Approaches

**Theory** (Markov chain Monte Carlo with simulated annealing)

- Jerrum-Sinclair-Vigoda '01: approximate permanent in $O^*(n^{10})$, yields $O^*((mn)^{10})$ algorithm for m x n binary contingency tables
- Bezáková-Bhatnagar-Vigoda '06: $O^*((mn)^3(m+n)^5)$

**Practice** (sequential importance sampling, Chen-Diaconis-Holmes-Liu '05)

- Bezáková-Sinclair-Štefankovič-Vigoda '06: negative example
- Jose Blanchet '06: SIS works if marginals $O(n^{1/4})$
- Bayati-Kim-Saberi '07: alternative importance sampling method, works if marginals $O(n^{1/4})$

**Practice** (the switching Markov chain, Diaconis-Gangolli '94)

- Kannan-Tetali-Vempala '97, Cooper-Dyer-Greenhill '05: works for regular marginals

# Different Approaches

**Theory** (Markov chain Monte Carlo with simulated annealing)

- Jerrum-Sinclair-Vigoda '01: approximate permanent in $O^*(n^{10})$, yields $O^*((mn)^{10})$ algorithm for m x n binary contingency tables
- Bezáková-Bhatnagar-Vigoda '06: $O^*((mn)^3(m+n)^5)$

**Practice** (sequential importance sampling, Chen-Diaconis-Holmes-Liu '05)

- Bezáková-Sinclair-Štefankovič-Vigoda '06: negative example
- Jose Blanchet '06: SIS works if marginals $O(n^{1/4})$
- Bayati-Kim-Saberi '07: alternative importance sampling method, works if marginals $O(n^{1/4})$

**Practice** (the switching Markov chain, Diaconis-Gangolli '94)

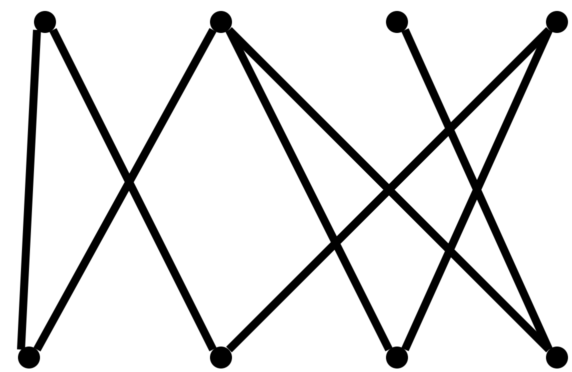
- Kannan-Tetali-Vempala '97, Cooper-Dyer-Greenhill '05: works for regular marginals

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



Perfect matching:

subset of vertex-disjoint edges covering all vertices

Permanent:

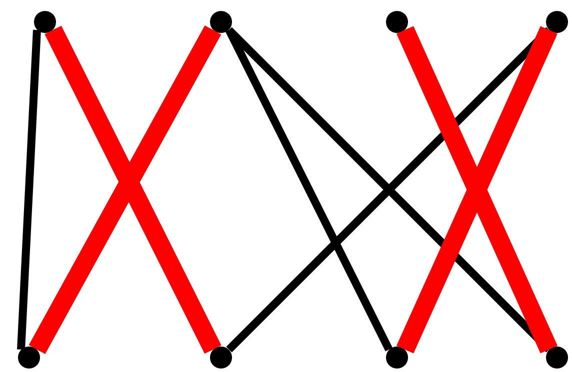
number of all perfect matchings

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



Perfect matching:

subset of vertex-disjoint edges covering all vertices

Permanent:

number of all perfect matchings

# Permanent: Broder Chain

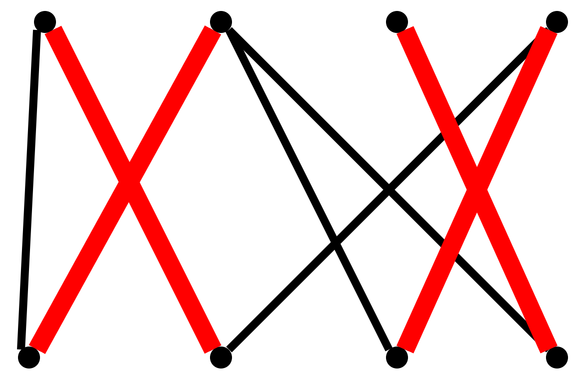
[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



# Permanent: Broder Chain

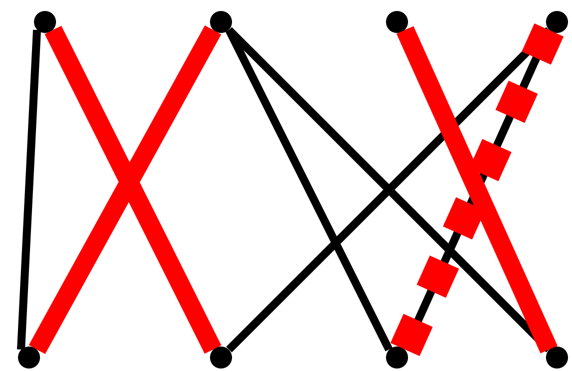
[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



# Permanent: Broder Chain

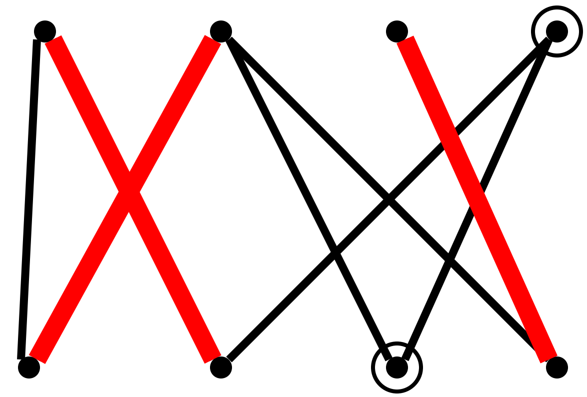
[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

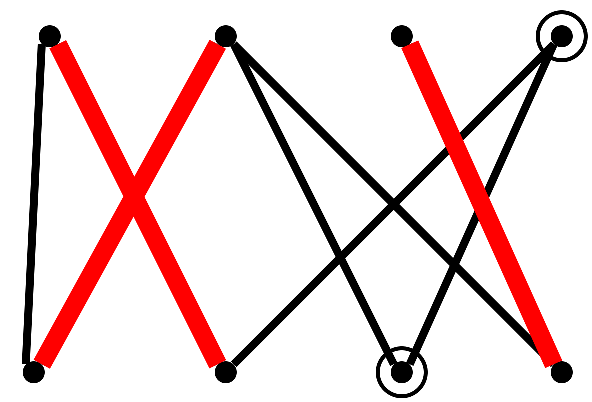
**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex

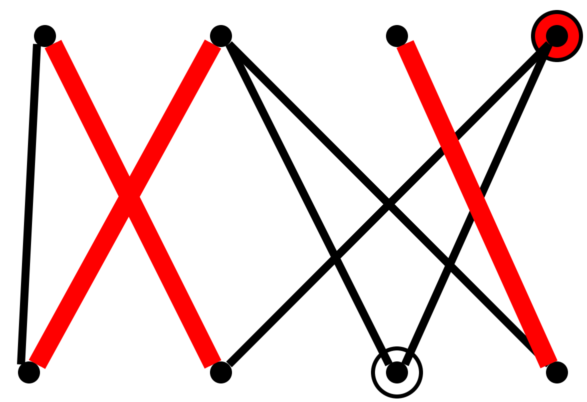


# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

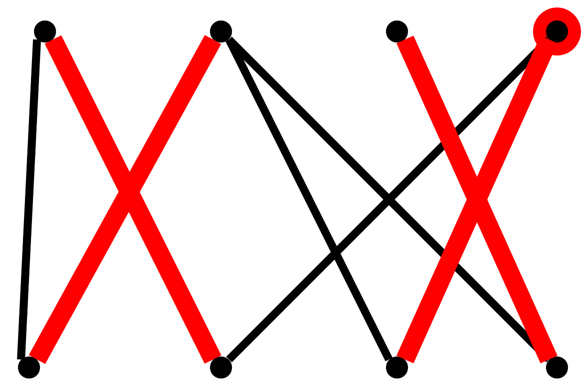
- choose a random vertex
    - if unmatched: match to the other hole

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

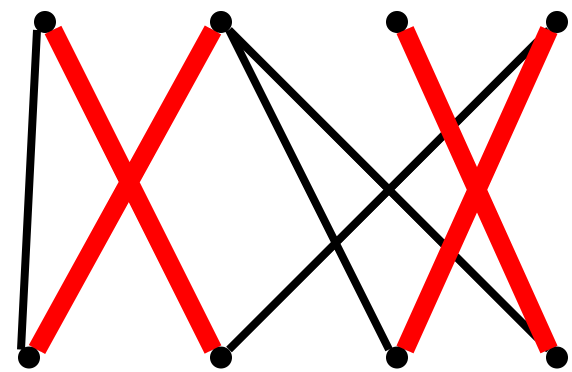
- choose a random vertex
    - if unmatched: match to the other hole

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

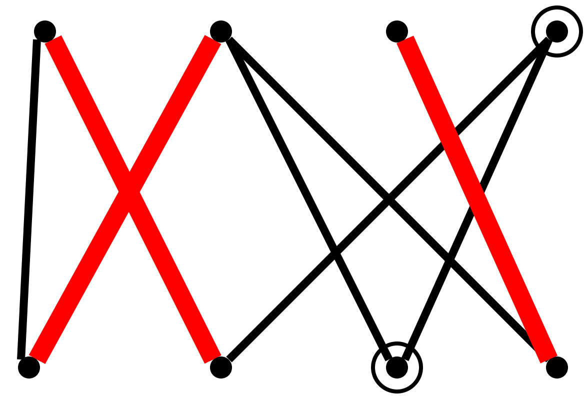
- choose a random vertex
    - if unmatched: match to the other hole

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
  - if unmatched: match to the other hole

# Permanent: Broder Chain

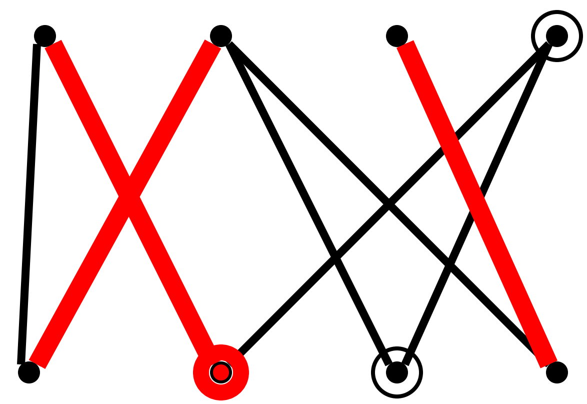
[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge



At a near matching:

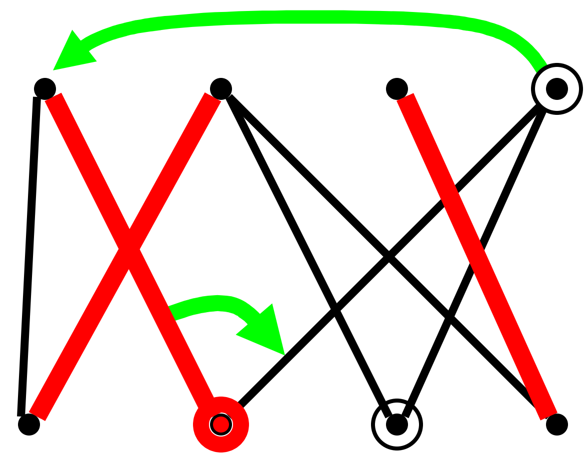
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

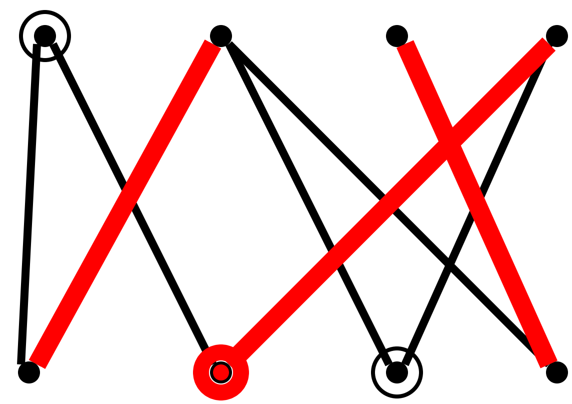
- choose a random vertex
  - if unmatched: match to the other hole
  - if matched: slide adjacent edge

# Permanent: Broder Chain

[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings



At a perfect matching:

- remove a random edge

At a near matching:

- choose a random vertex
    - if unmatched: match to the other hole
    - if matched: slide adjacent edge

# Permanent: Broder Chain

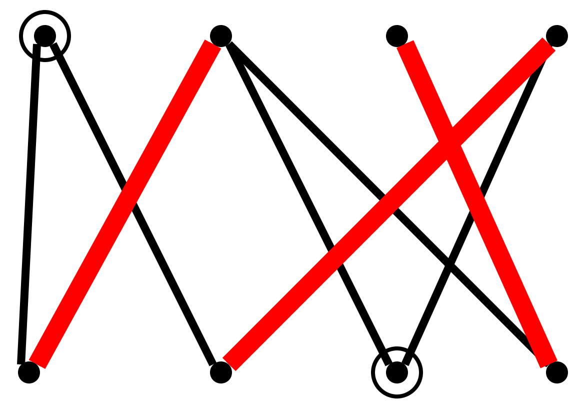
[Broder '88]

**What for:** uniform sampling of perfect matchings

**How:** Markov chain on perfect + near-perfect matchings

At a perfect matching:

- remove a random edge

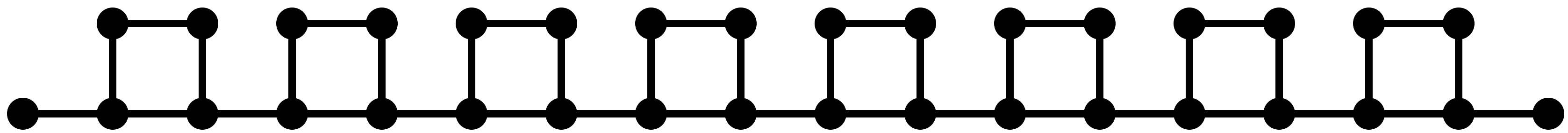


At a near matching:

- choose a random vertex
    - if unmatched: match to the other hole
    - if matched: slide adjacent edge

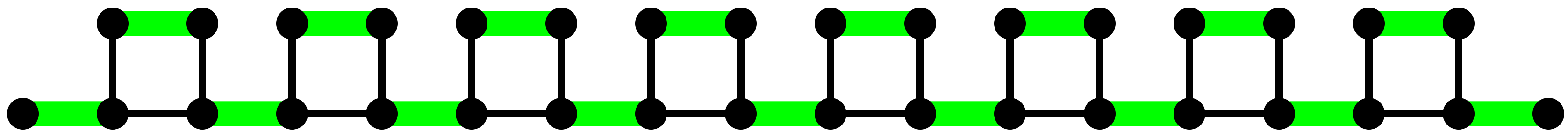
# Broder Chain

**Mixes in polynomial time?** Even if it did…



## Broder Chain

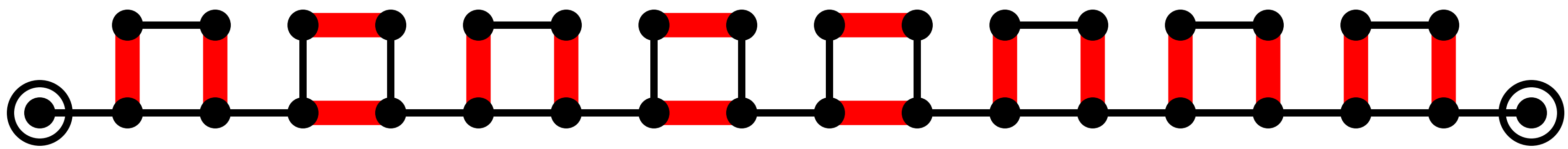
**Mixes in polynomial time?** Even if it did...



1 perfect matching

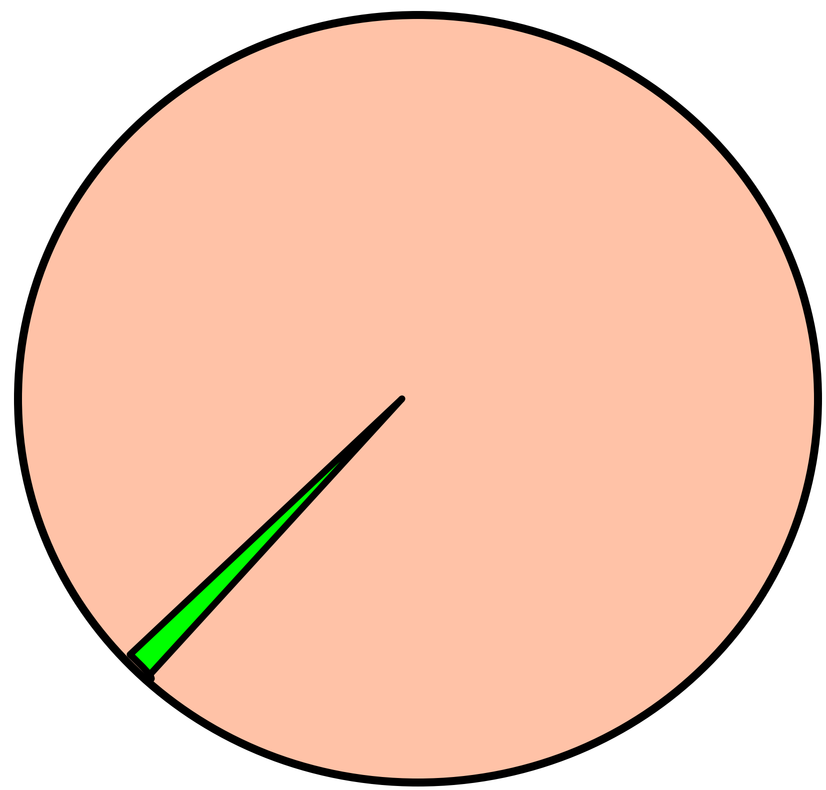
# Broder Chain

**Mixes in polynomial time?** Even if it did…



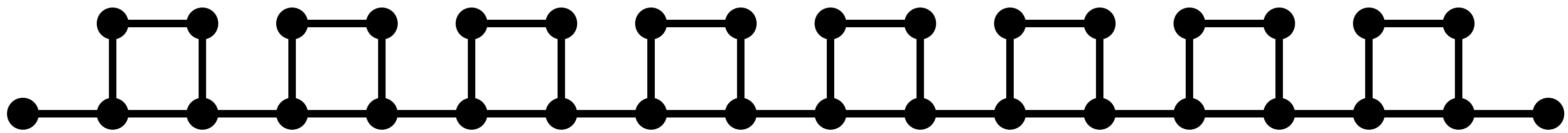
1 perfect matching

$\geq 2^{(n/4)}$ near matchings



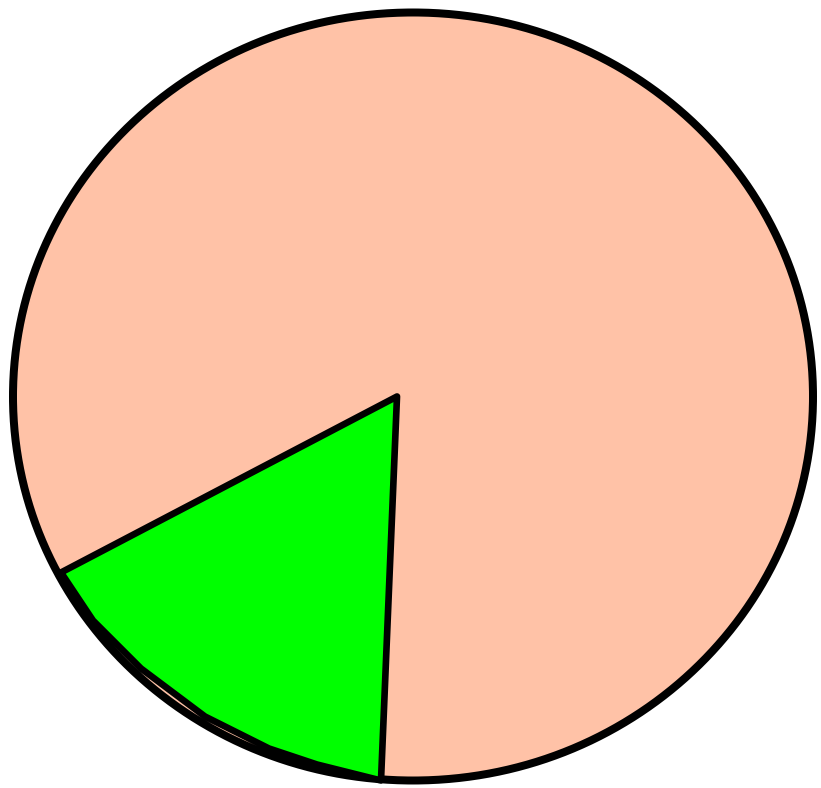
# Broder Chain

**Mixes in polynomial time?** Even if it did...



1 perfect matching

$\geq 2^{(n/4)}$ near matchings



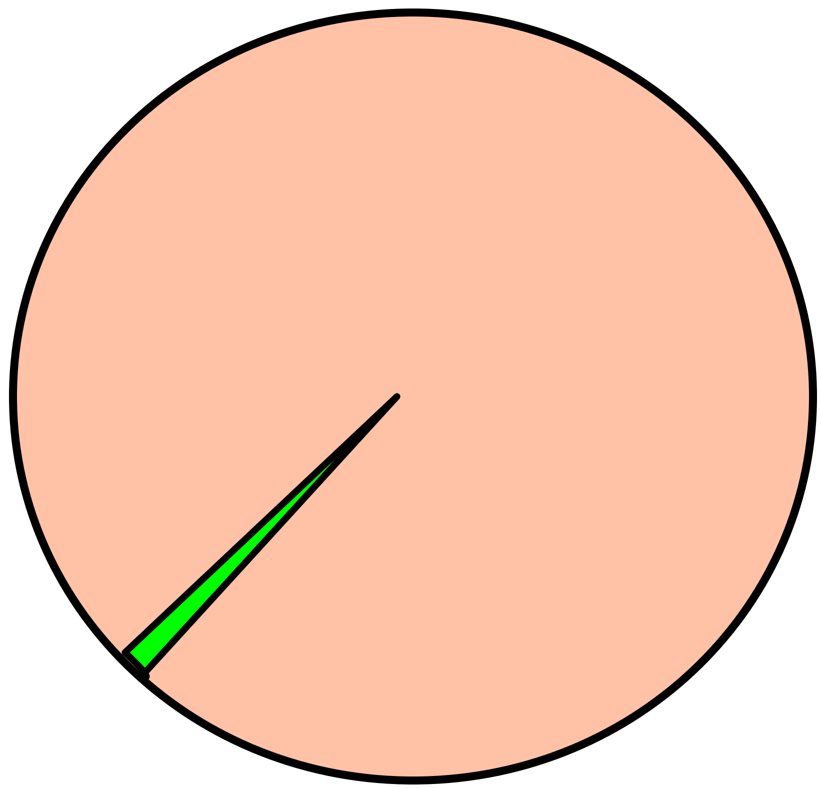
**Thm** [Jerrum-Sinclair '89]:

Rapid mixing if perfects **polynomially** related to nears.

# Simulated Annealing for Permanent

Jerrum-Sinclair-Vigoda '01:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

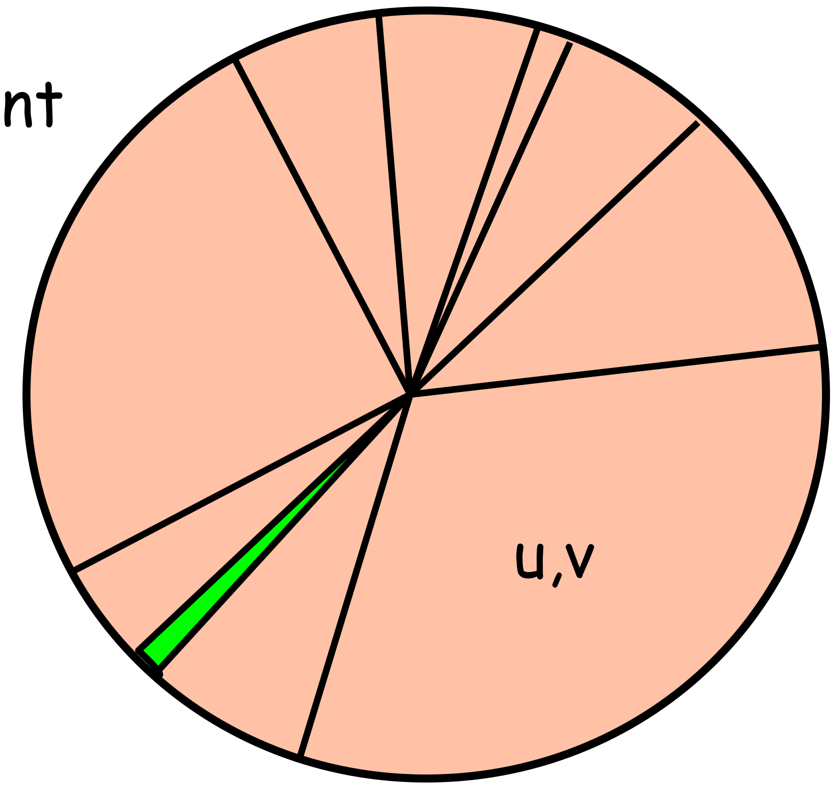


# Simulated Annealing for Permanent

Jerrum-Sinclair-Vigoda '01:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

Originally: $n^2$+1 regions, very different size

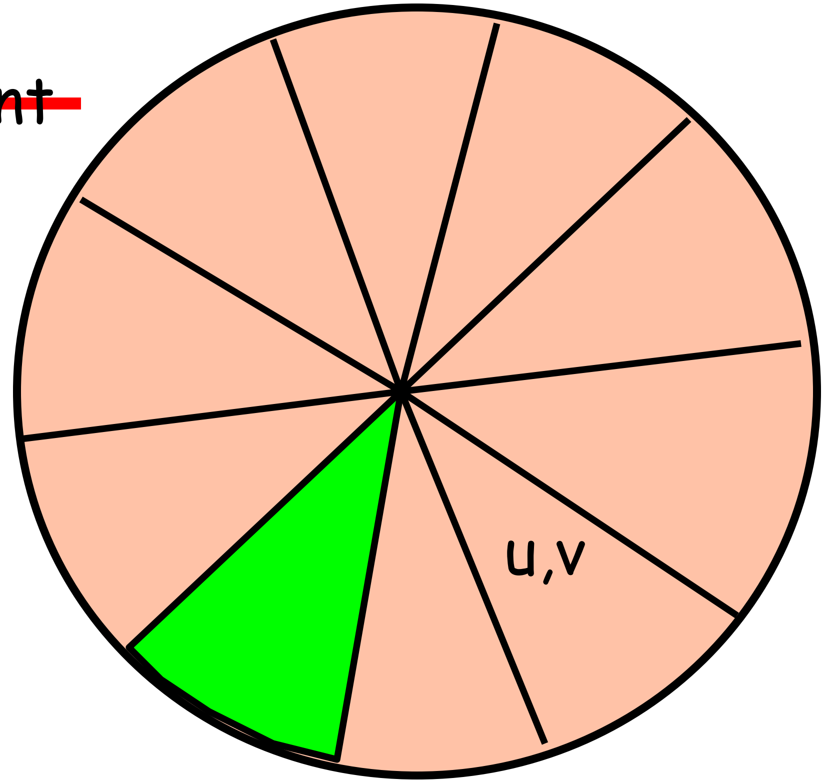


# Simulated Annealing for Permanent

Jerrum-Sinclair-Vigoda '01:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

Want: $n^2$+1 regions, ~~very different~~ the same size

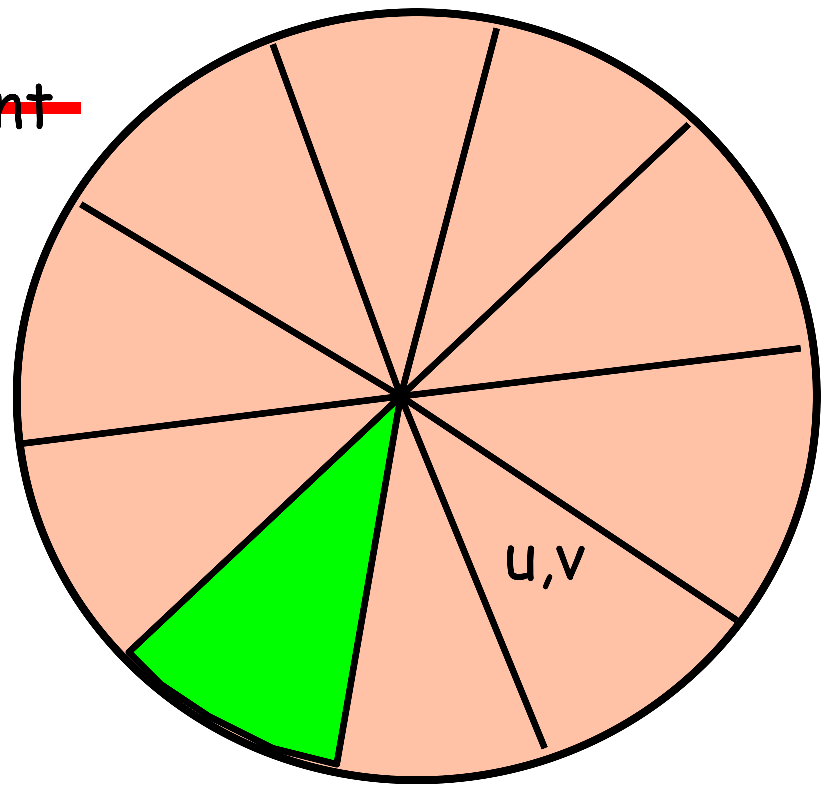


# Simulated Annealing for Permanent

Jerrum-Sinclair-Vigoda '01:

Change the **weight** so that **perfect matchings** take **polynomial** fraction.

Want: $n^2$+1 regions,
~~very different~~
the same size




**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

# Simulated Annealing for Permanent

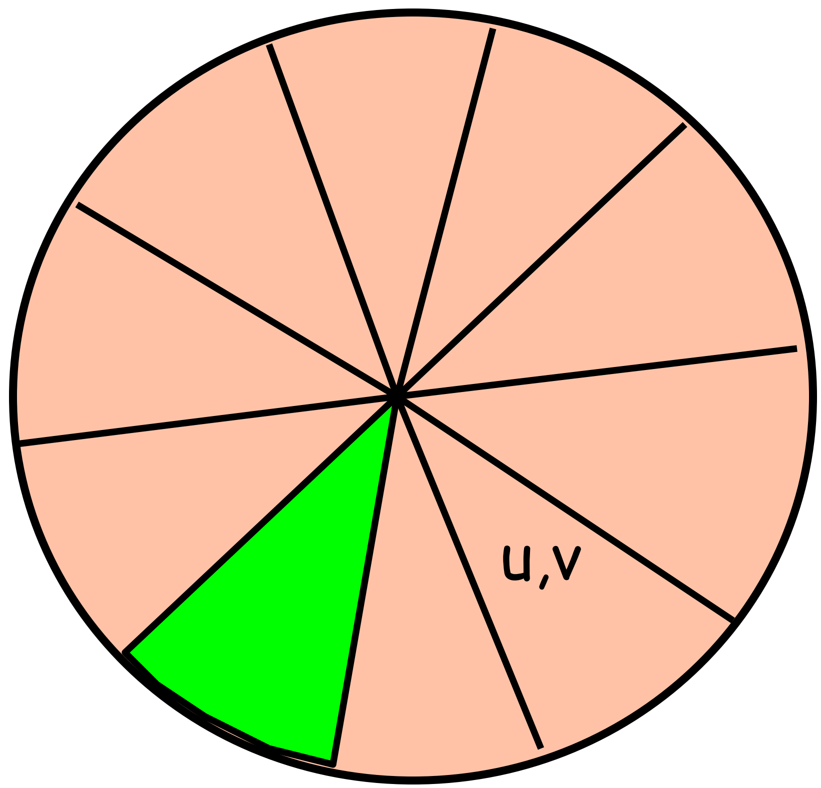
**Good:** A perfect matching sampled with prob. $1/(n^2+1)$

**Bad:** Computing ideal weights as hard as original problem **?**

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

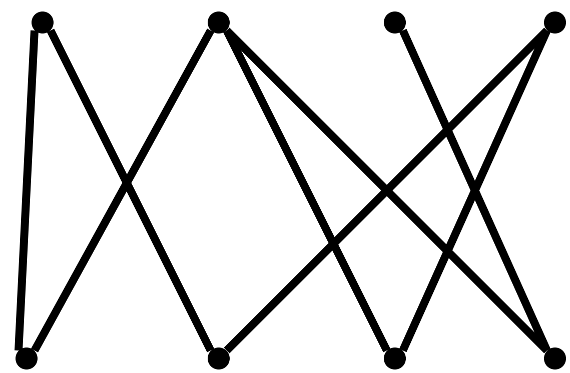
(EASY) (DIFFICULT)

G

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



G

# Simulated Annealing for Permanent

Unordered state → Ordered state

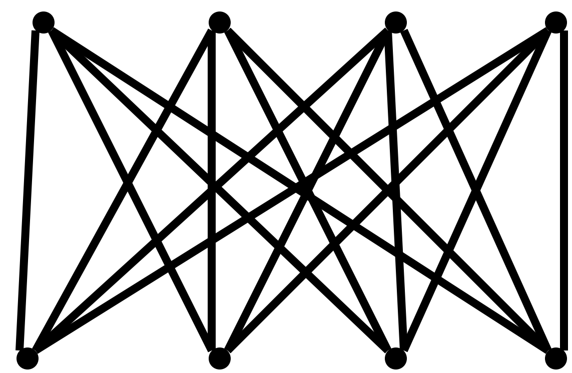
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



$K_{n,n}$

# Simulated Annealing for Permanent

Unordered state → Ordered state

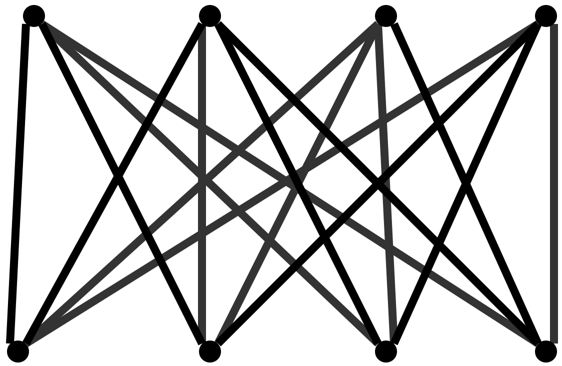
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\textbf{(\# perfects)}}{\textbf{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

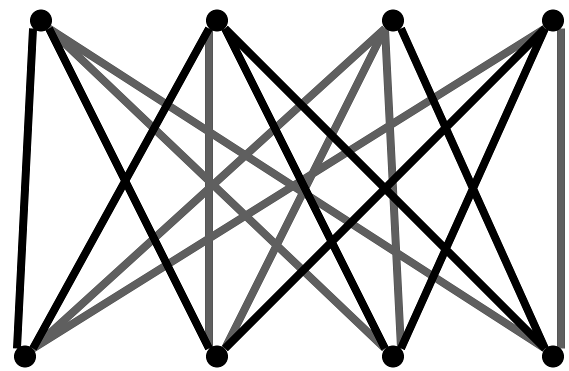
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

(EASY) (DIFFICULT)

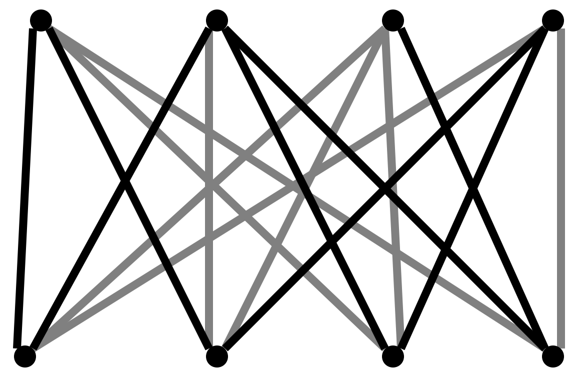
$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**

(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

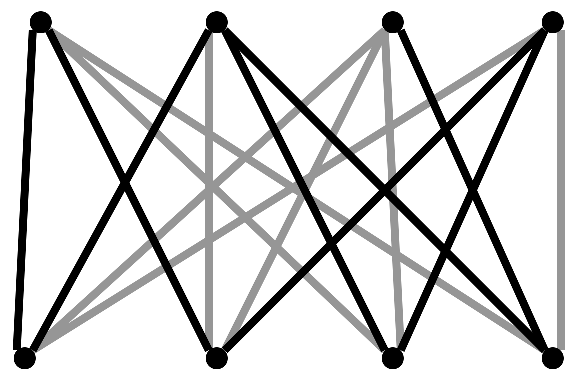
(EASY) (DIFFICULT)

$K_{n,n} \longrightarrow G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes } u,v\text{)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

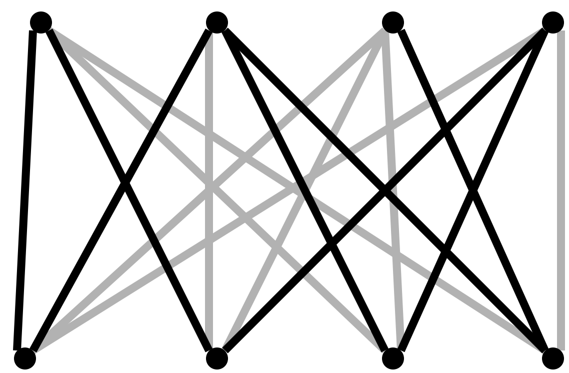
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

(EASY) (DIFFICULT)

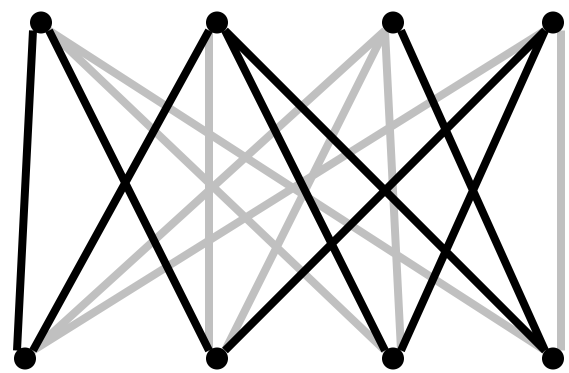
$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

**(# perfects)**

**(# nears with holes u,v)**



# Simulated Annealing for Permanent

Unordered state → Ordered state

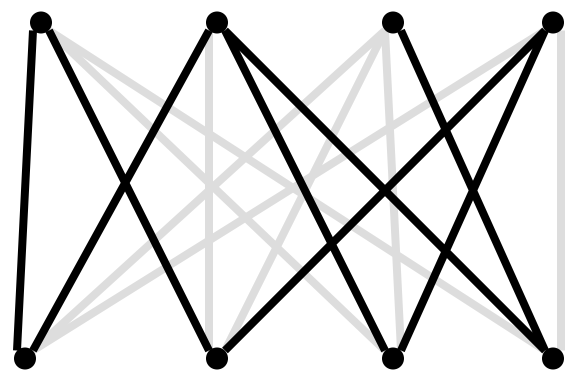
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

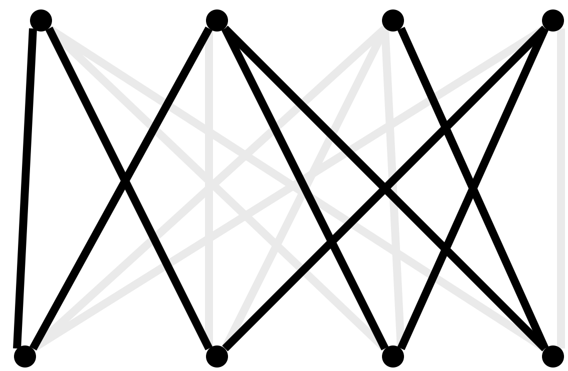
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Unordered state → Ordered state

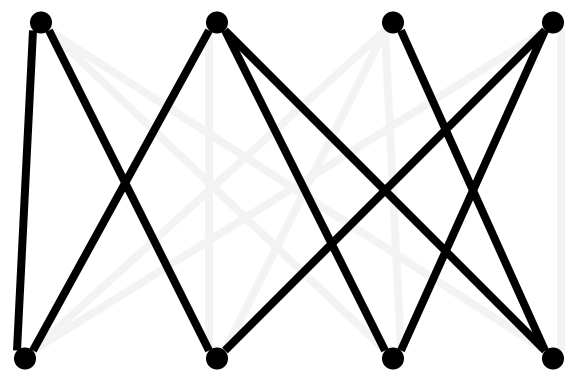
(EASY) (DIFFICULT)

$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**
(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



G

# Simulated Annealing for Permanent

Unordered state → Ordered state

(EASY) (DIFFICULT)

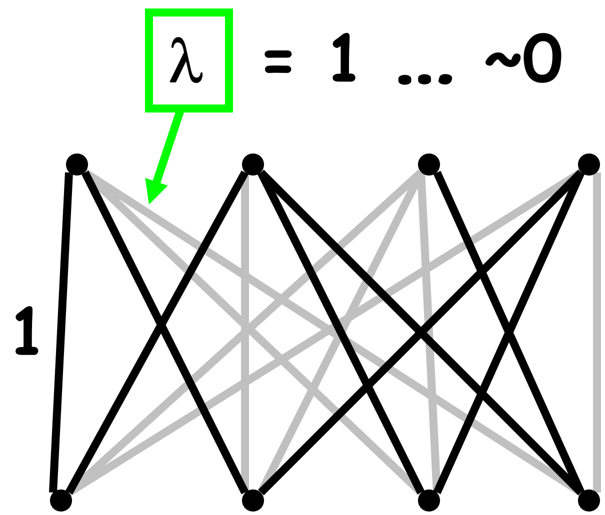
$K_{n,n}$ → $G$

Solution: **Approximate**

**Ideal weights**

(for a matching with holes u,v):

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$



# Simulated Annealing for Permanent

Ideal weights $\dfrac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$

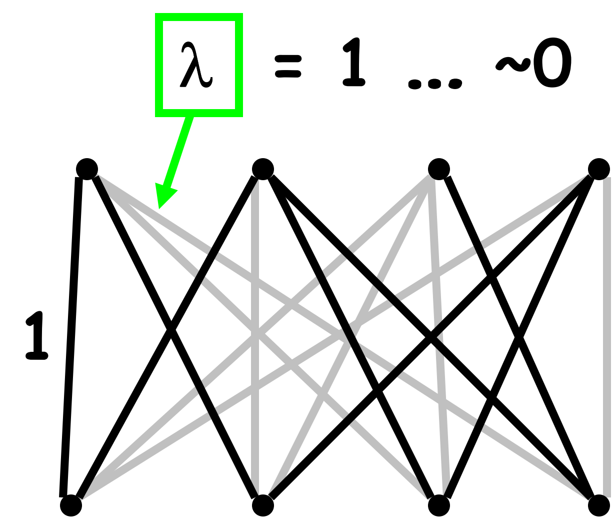
need to be λ-weighted:

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

where

$$\lambda(M) = \lambda^{\#\ \lambda\text{-edges in } M}$$

$$\lambda(S) = \sum_{M \text{ in } S} \lambda(M)$$



# Simulated Annealing for Permanent

**Thm** [Jerrum-Sinclair-Vigoda '01]:
Weighted Broder chain mixes if w(u,v) approximated within a constant factor.

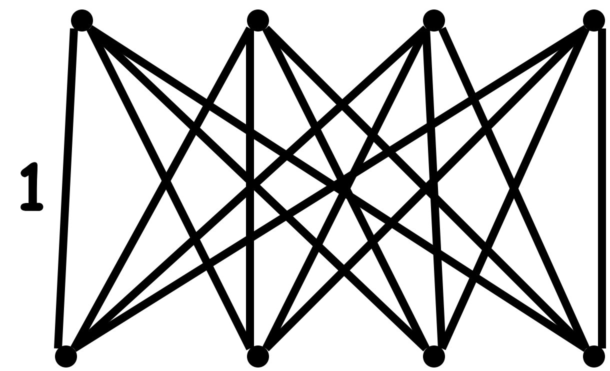
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Algorithm** (sketch):

* Initially, $\lambda = 1$.
Thus $w(u,v) = n!/(n-1)! = n$.

Later, have approx. of w(u,v).
Run chain to improve the approx.
Decrease $\lambda$ (until ~0).

$\lambda = 1$




(Improved approx. of old $\lambda$ = starting approx. of new $\lambda$)

# Simulated Annealing for Permanent

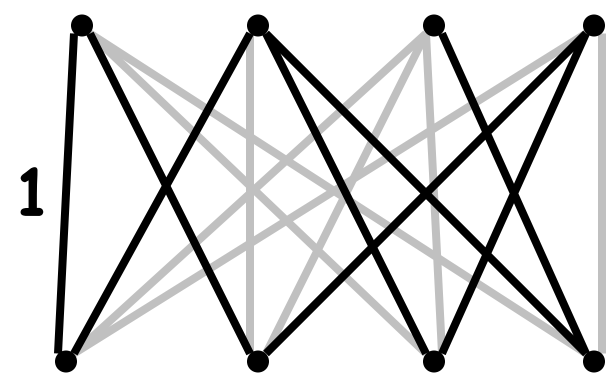
**Thm** [Jerrum-Sinclair-Vigoda '01]:
Weighted Broder chain mixes if $w(u,v)$ approximated within a constant factor.

$$\boxed{\underbrace{w(u,v)}_{\text{4-apx}} = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}}$$

**Algorithm** (sketch):

Initially, $\lambda = 1$.
Thus $w(u,v) = n!/(n-1)! = n$.

$\lambda = 0.7$




* Later, have approx. of $w(u,v)$.
Run chain to improve the approx.
Decrease $\lambda$ (until ~0).

(Improved approx. of old $\lambda$ = starting approx. of new $\lambda$)

# Simulated Annealing for Permanent

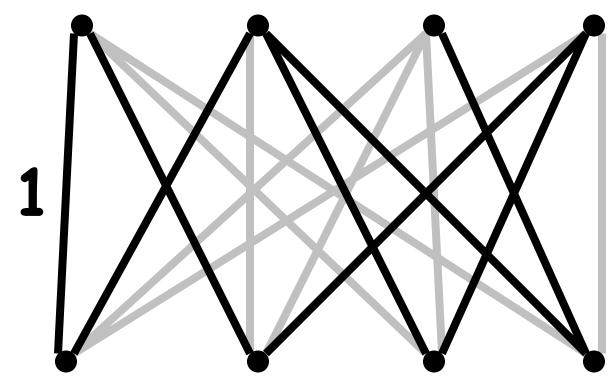
**Thm** [Jerrum-Sinclair-Vigoda '01]:
Weighted Broder chain mixes if w(u,v) approximated within a constant factor.

$$\boxed{w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}}$$

~~4~~ 2-apx

**Algorithm** (sketch):

$\lambda = 0.7$




Initially, $\lambda = 1$.
Thus $w(u,v) = n!/(n-1)! = n$.

Later, have approx. of w(u,v).
* Run chain to improve the approx.
Decrease $\lambda$ (until ~0).

(Improved approx. of old $\lambda$ = starting approx. of new $\lambda$)

# Simulated Annealing for Permanent

**Thm** [Jerrum-Sinclair-Vigoda '01]:
Weighted Broder chain mixes if w(u,v) approximated within a constant factor.

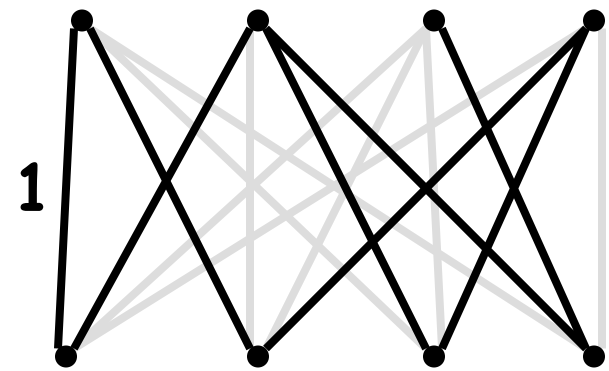
$$w(u,v) \;=\; \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

~~4~~-apx
2 = 4-apx for
$\lambda$ = ~~0.7~~ 0.6

**Algorithm** (sketch):

Initially, $\lambda = 1$.
Thus $w(u,v) = n!/(n-1)! = n$.

Later, have approx. of w(u,v).
Run chain to improve the approx.
* Decrease $\lambda$ (until ~0).




(Improved approx. of old $\lambda$ = starting approx. of new $\lambda$)

# Simulated Annealing for Permanent

**Thm** [Jerrum-Sinclair-Vigoda '01]:
Weighted Broder chain mixes if w(u,v) approximated within a constant factor.

$$\boxed{w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}}$$

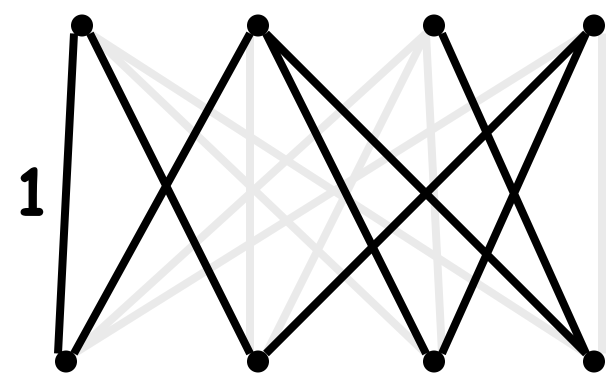
$\cancel{4}$ 2-apx = 4-apx for $\lambda$ = $\cancel{0.7}$ $\cancel{0.6}$ …~0

**Algorithm** (sketch):

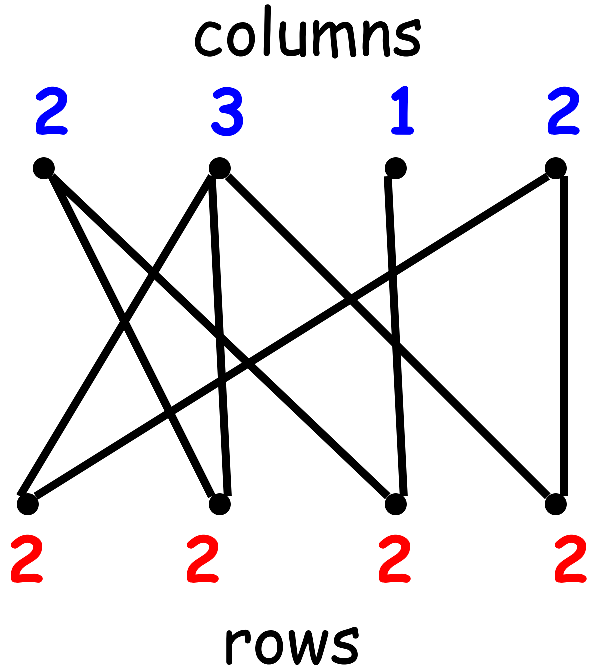
Initially, $\lambda = 1$.
Thus $w(u,v) = n!/(n-1)! = n$.

Later, have approx. of w(u,v).
Run chain to improve the approx.
* Decrease $\lambda$ (until ~0).

(Improved approx. of old $\lambda$ = starting approx. of new $\lambda$)



# BCT: Bipartite Graphs with Given Degrees

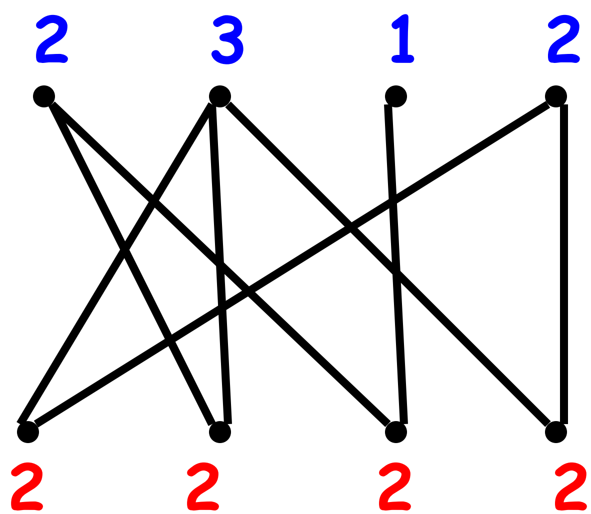



columns

rows

$$\begin{array}{cc} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} 2 \\ 2 \\ 2 \\ 2 \end{array} \\ \begin{array}{cccc} 2 & 3 & 1 & 2 \end{array} & \end{array}$$

# BCT: Bipartite Graphs with Given Degrees




columns

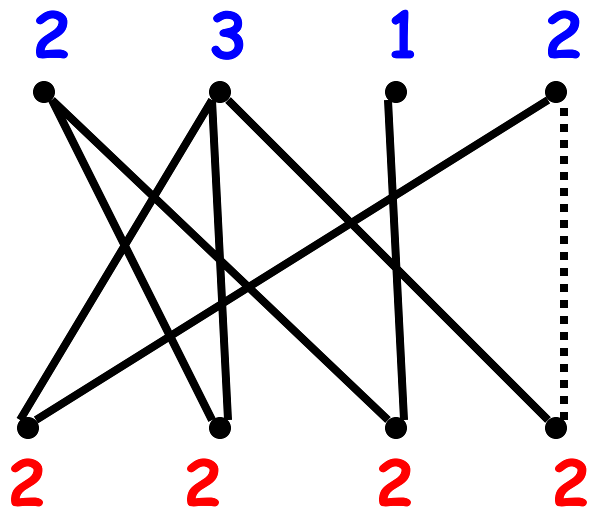
| rows | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 2 |
| | 1 | 1 | 0 | 0 | 2 |
| | 1 | 0 | 1 | 0 | 2 |
| | 0 | 1 | 0 | 1 | 2 |
| | 2 | 3 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

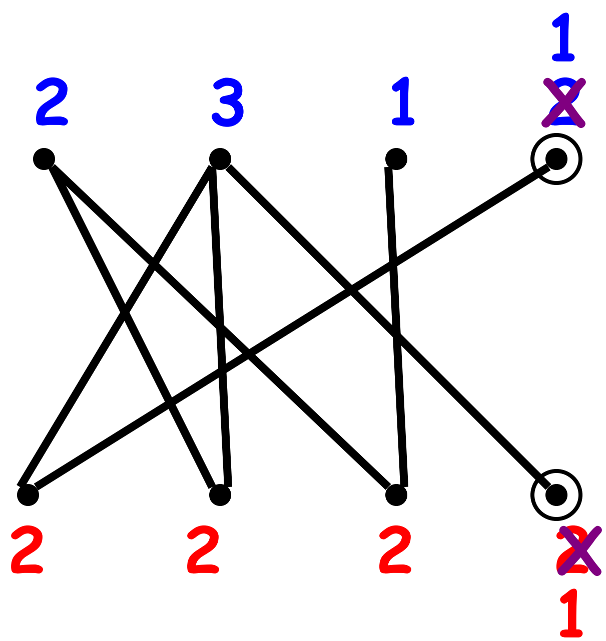
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 2 | 3 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

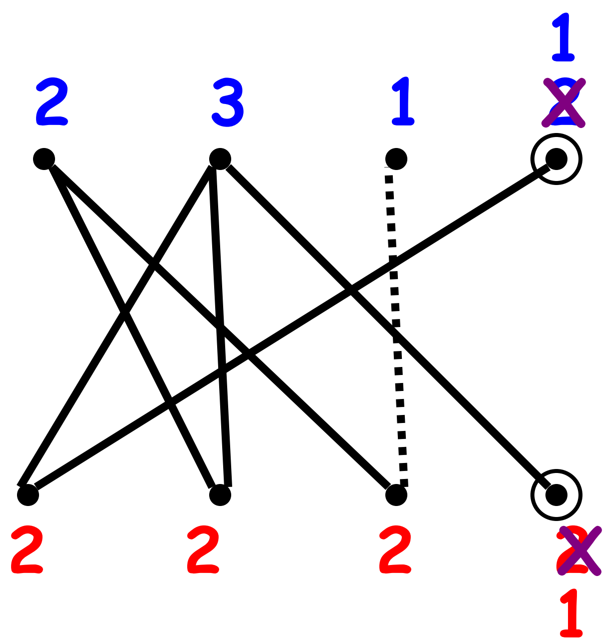
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | 1 | ~~2~~ 1 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

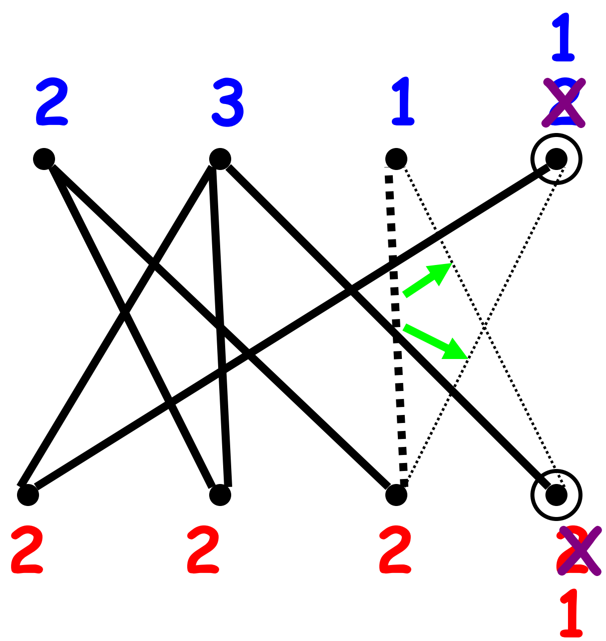
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | 1 | ~~2~~ 1 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

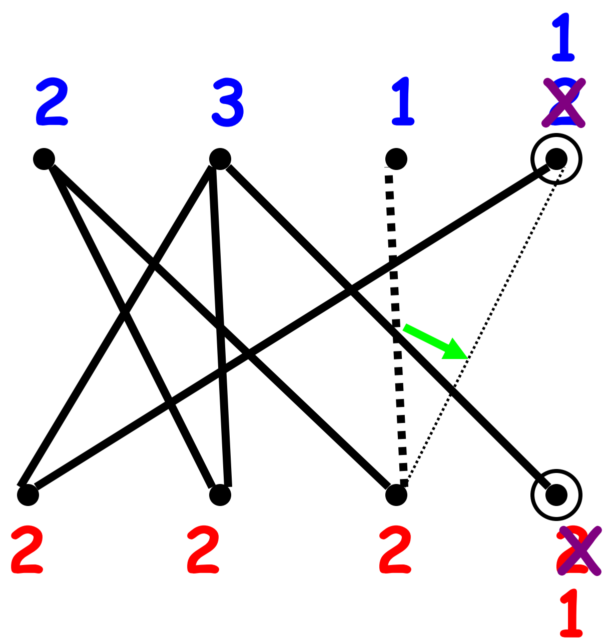
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | 1 | ~~2~~ 1 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

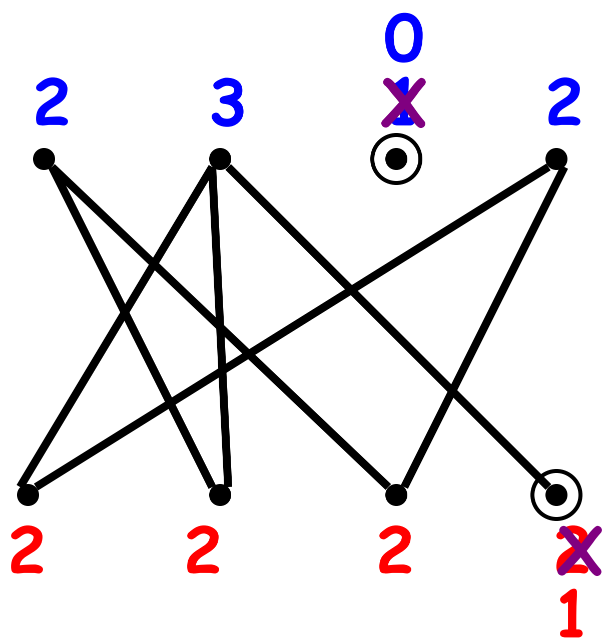
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | 1 | ~~2~~ 1 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

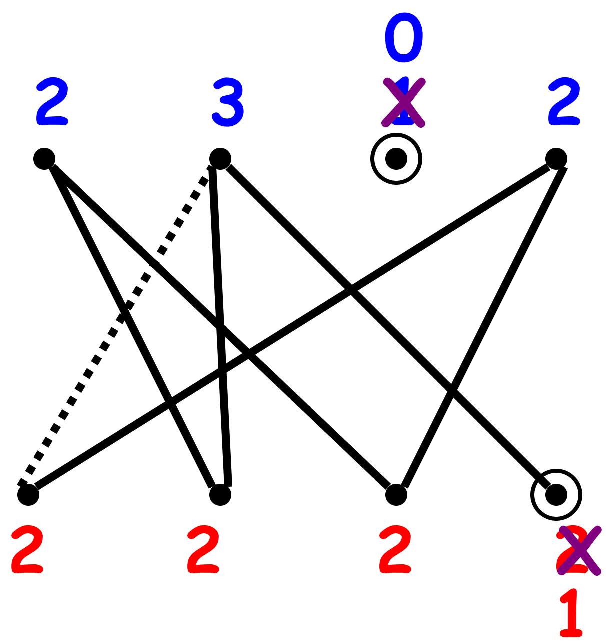
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | ~~1~~ 0 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

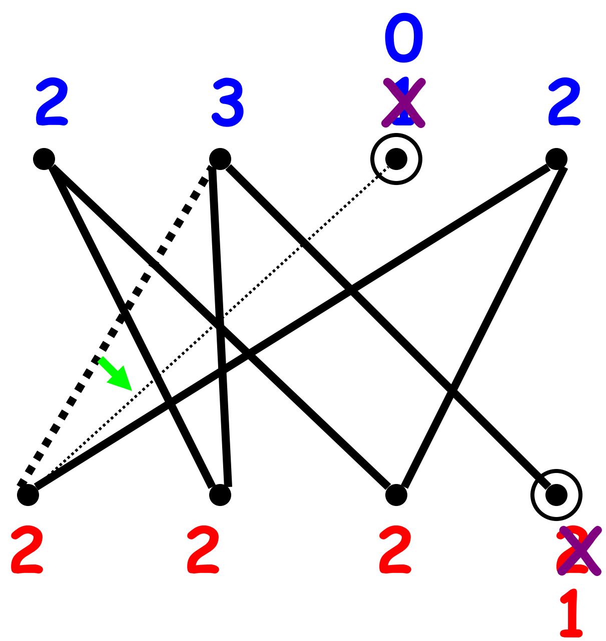
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | 3 | ~~1~~ 0 | 2 | |

**"Sliding" Markov Chain** on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

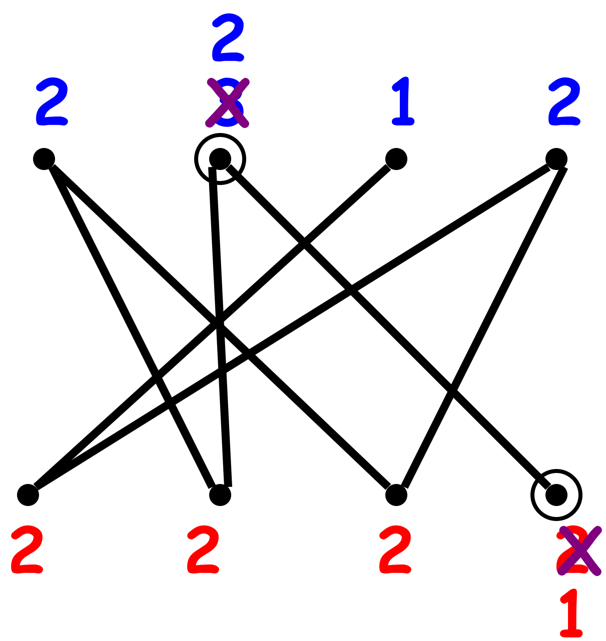
| | | | | | |
|---|---|---|---|---|---|
| rows | 0 | 1 | 0 | 1 | 2 |
| | 1 | 1 | 0 | 0 | 2 |
| | 1 | 0 | 0 | 1 | 2 |
| | 0 | 1 | 0 | 0 | ~~2~~ 1 |
| | 2 | 3 | ~~1~~ 0 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

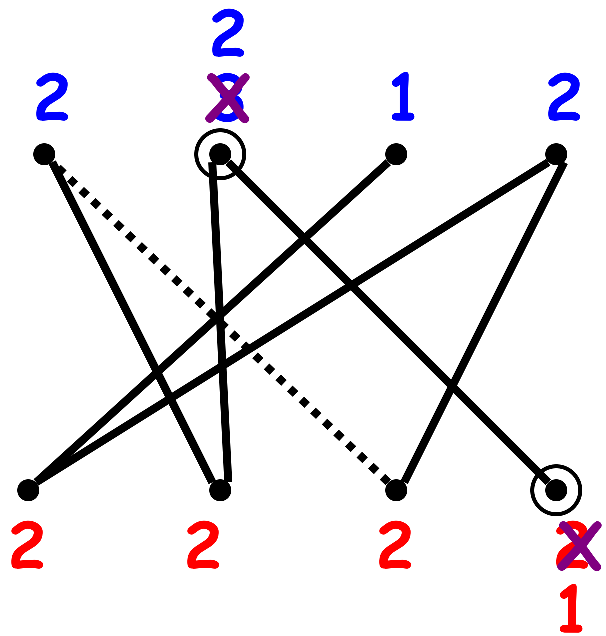
| rows | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 2 |
| | 1 | 0 | 1 | 0 | 2 |
| | 1 | 0 | 0 | 1 | 2 |
| | 0 | 1 | 0 | 0 | ~~2~~ 1 |
| | 2 | ~~3~~ 2 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

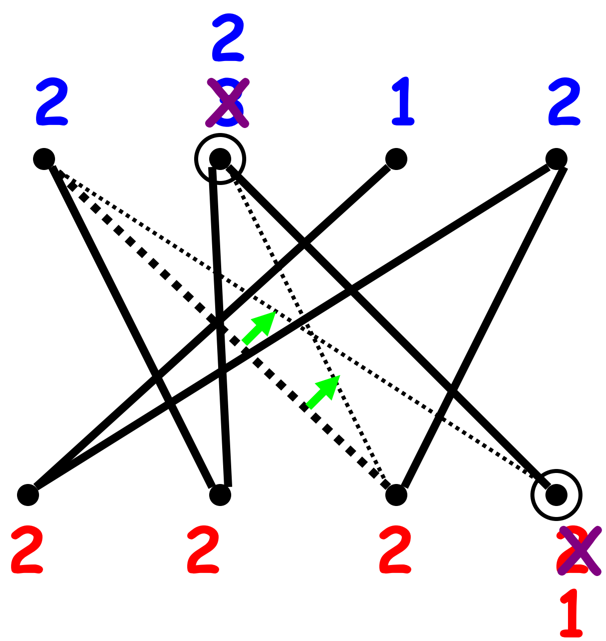
| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | ~~3~~ 2 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 0 | ~~2~~ 1 |
| 2 | ~~3~~ 2 | 1 | 2 | |

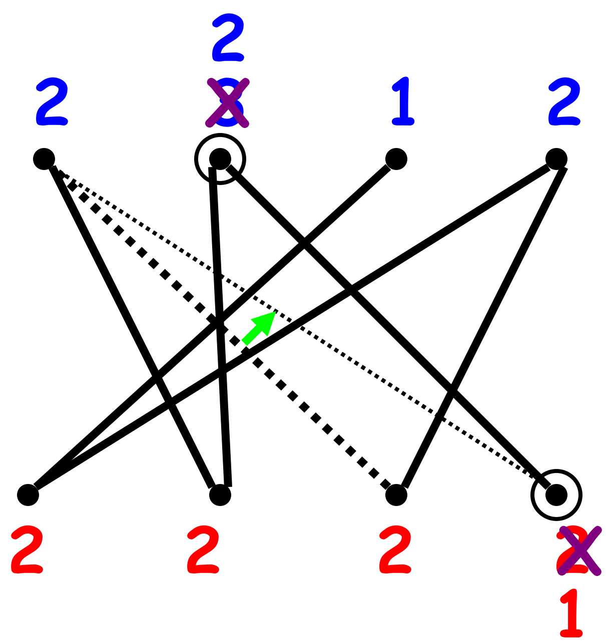
## "Sliding" Markov Chain

on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

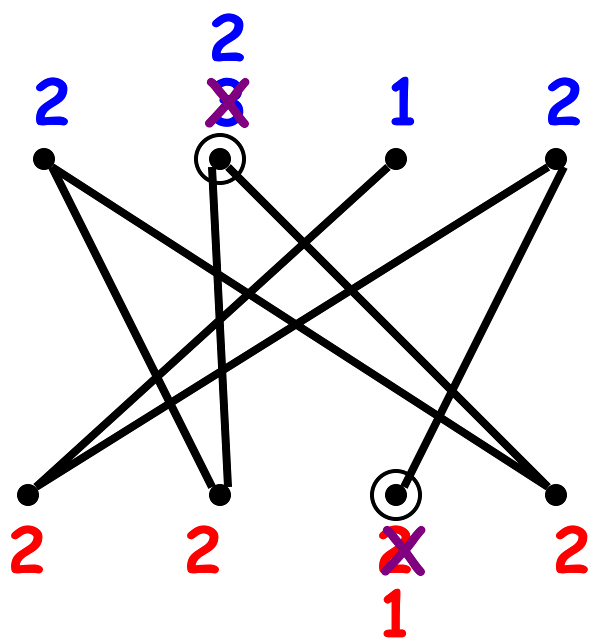
| rows | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 2 |
| | 1 | 0 | 1 | 0 | 2 |
| | 1 | 0 | 0 | 1 | 2 |
| | 0 | 1 | 0 | 0 | ~~2~~ 1 |
| | 2 | ~~3~~ 2 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees



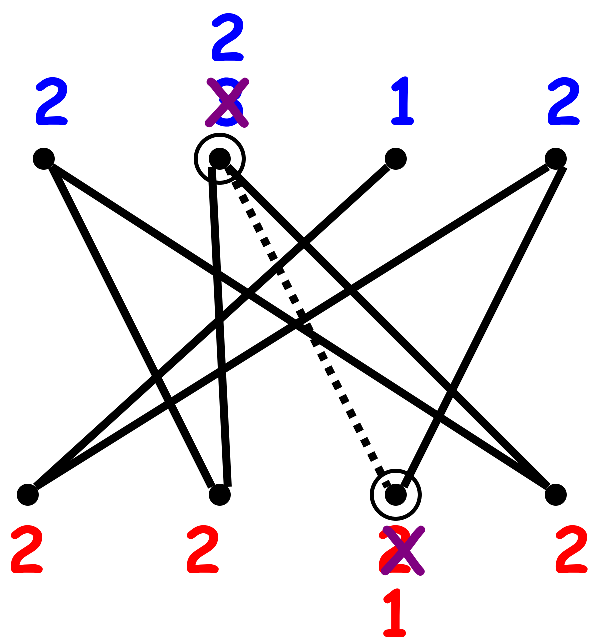

| | columns | | | | |
|---|---|---|---|---|---|
| rows | 0 | 1 | 0 | 1 | 2 |
| | 1 | 0 | 1 | 0 | 2 |
| | 0 | 0 | 0 | 1 | ~~2~~ 1 |
| | 1 | 1 | 0 | 0 | 2 |
| | 2 | ~~3~~ 2 | 1 | 2 | |

**"Sliding" Markov Chain** on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

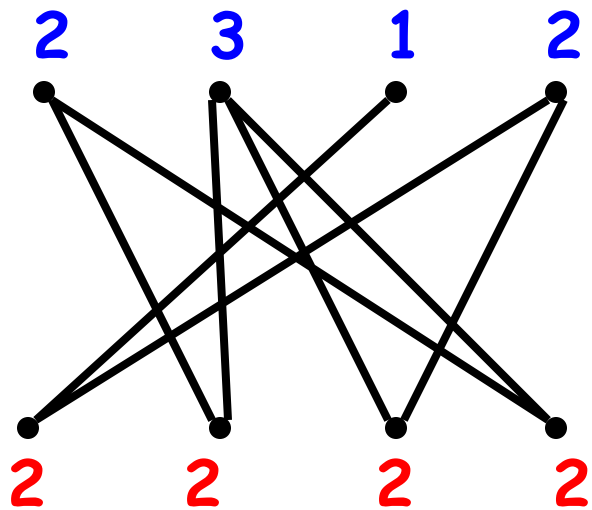
| rows | | | | | |
|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 2 |
| | 1 | 0 | 1 | 0 | 2 |
| | 0 | 0 | 0 | 1 | ~~2~~ 1 |
| | 1 | 1 | 0 | 0 | 2 |
| | 2 | ~~3~~ 2 | 1 | 2 | |

## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# BCT: Bipartite Graphs with Given Degrees




columns

rows

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 2 |
| 1 | 0 | 1 | 0 | 2 |
| 0 | 1 | 0 | 1 | 2 |
| 1 | 1 | 0 | 0 | 2 |
| 2 | 3 | 1 | 2 | |

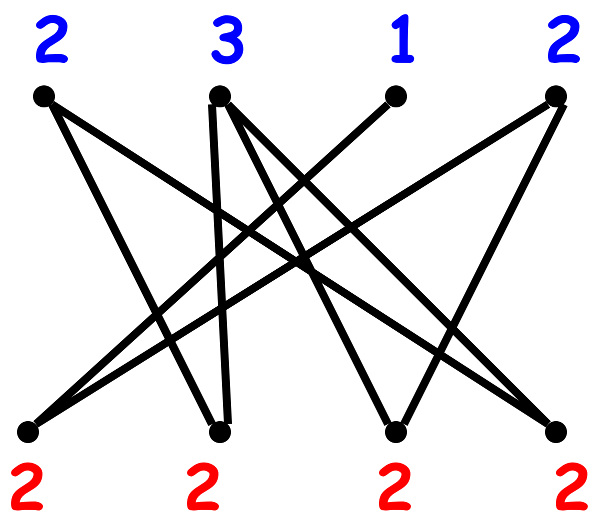
## "Sliding" Markov Chain on perfect and near tables

Perfect: remove a random edge

Near: slide edges or match

# Simulated Annealing for BCT ?

[ Bezáková-Bhatnagar-Vigoda '06 ]




Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

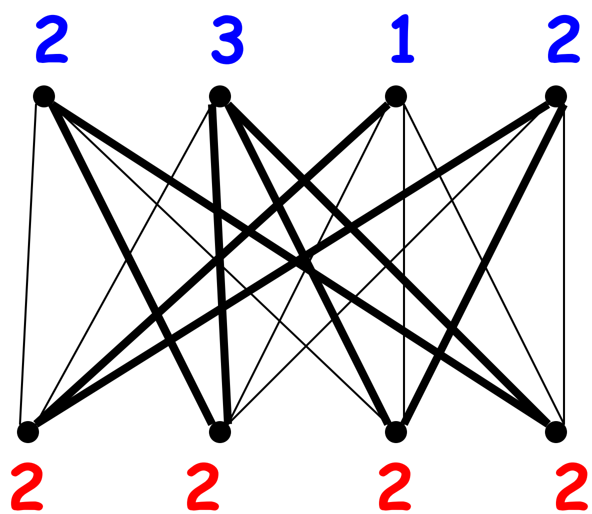
Unordered state → Ordered state

(EASY) (DIFFICULT)

? → 2 3 1 2 / 2 2 2 2

# Simulated Annealing for BCT ?

[ Bezáková-Bhatnagar-Vigoda '06 ]




Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

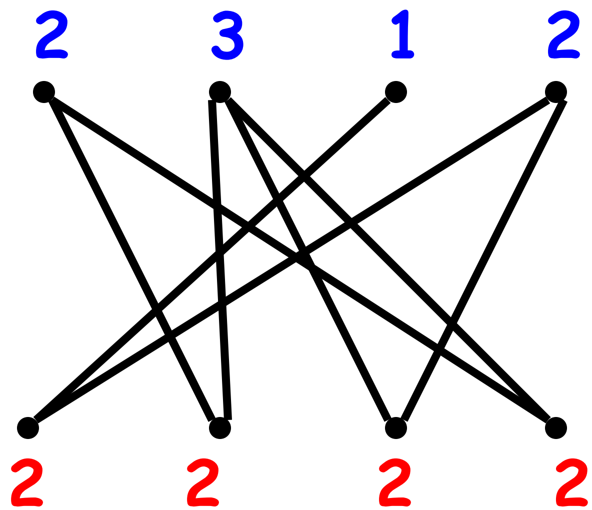
Unordered state → Ordered state

(EASY) (DIFFICULT)

? → 2 3 1 2 / 2 2 2 2 on $K_{n,n}$

# Simulated Annealing for BCT ?

[ Bezáková-Bhatnagar-Vigoda '06 ]

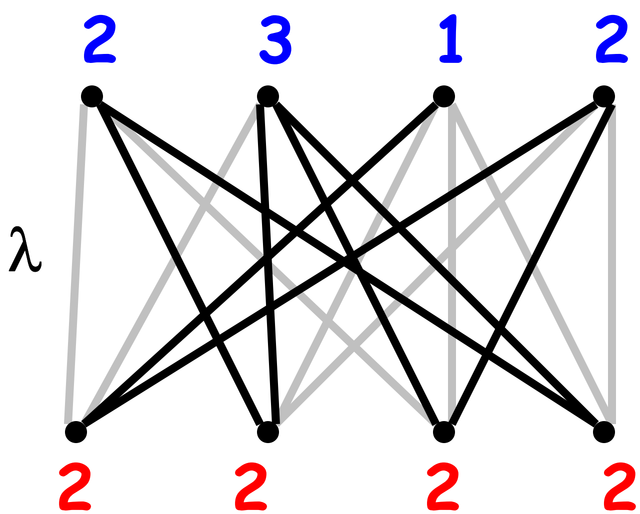



Ideal weights

$$\frac{\text{(\# perfects)}}{\text{(\# nears with holes u,v)}}$$

| Unordered state | → | Ordered state |
|---|---|---|
| (EASY) | | (DIFFICULT) |
| 2 3 1 2 <br> 2 2 2 2 on $G^*$ | → | 2 3 1 2 <br> 2 2 2 2 on $K_{n,n}$ |

# Simulated Annealing for BCT ?



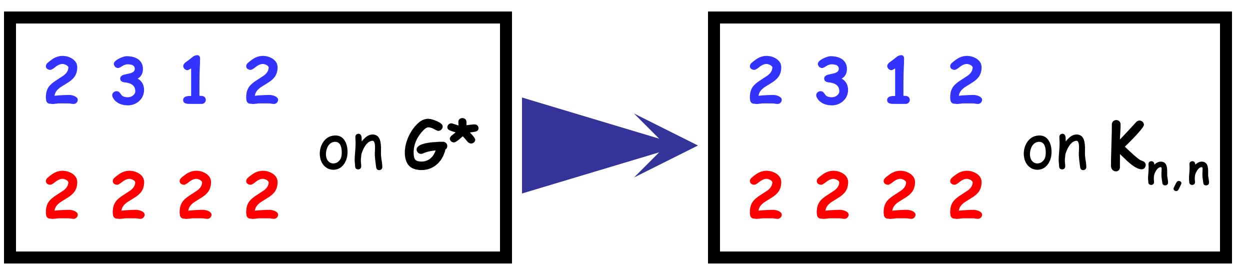

Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

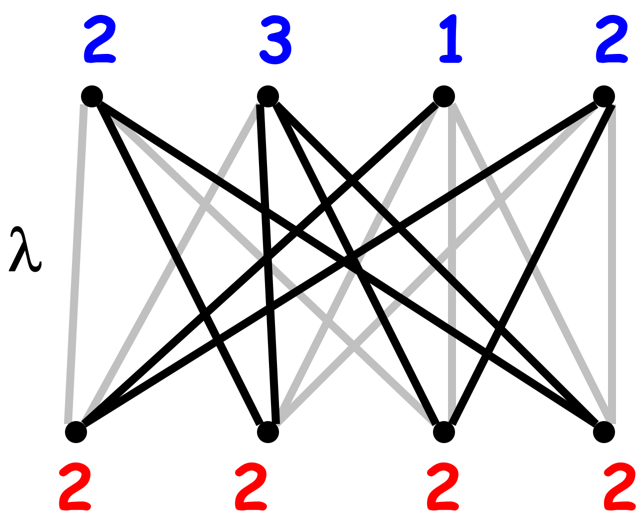
where $\lambda(T) = \lambda^{\#\ \lambda\text{-edges in } T}$

$$\lambda(S) = \sum_{T \text{ in } S} \lambda(T)$$

$\lambda$ = ~0 ........................ 1



# Simulated Annealing for BCT ?




Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

where $\lambda(T) = \lambda^{\#\,\lambda\text{-edges in } T}$

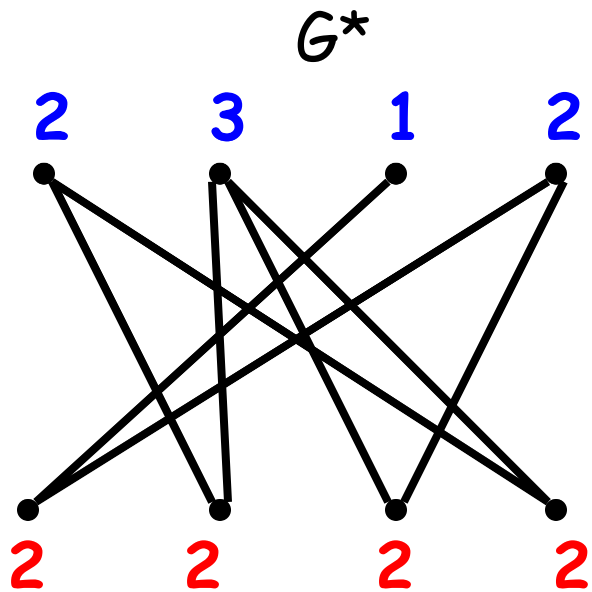
$$\lambda(S) = \sum_{T \text{ in } S} \lambda(T)$$

**The catch** :

What, if for some u,v, there is no near-table which uses all real edges ? Then,

$$\lambda(\mathscr{N}(u,v)) = 0 \quad \text{for } \lambda = 0.$$

# Simulated Annealing for BCT ?



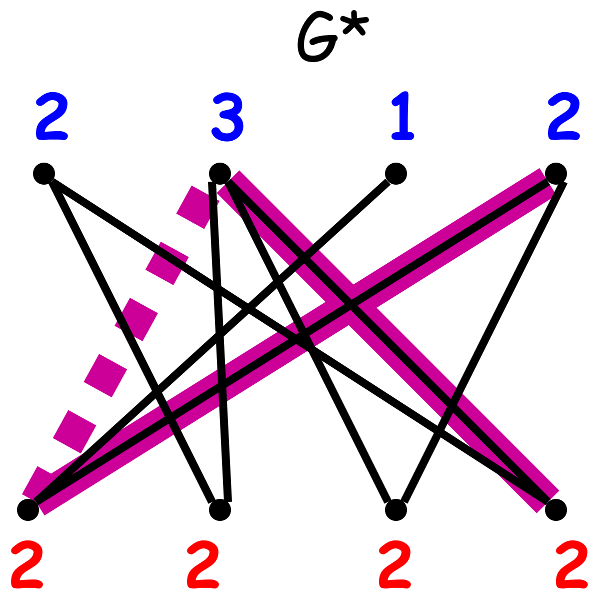

Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

# Simulated Annealing for BCT ?




Recall that

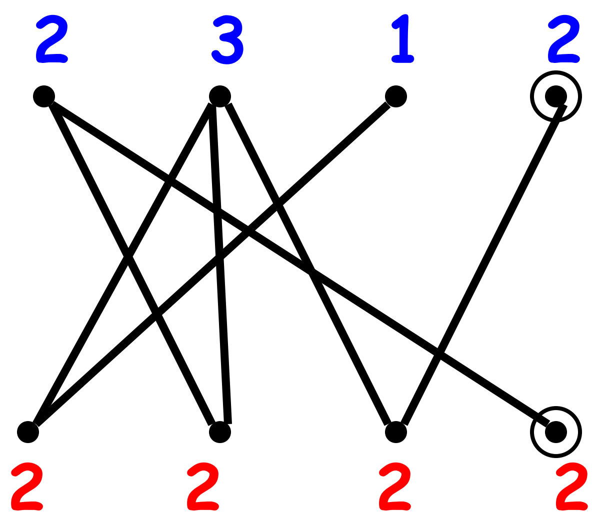
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

# Simulated Annealing for BCT ?

a near-table T




Recall that

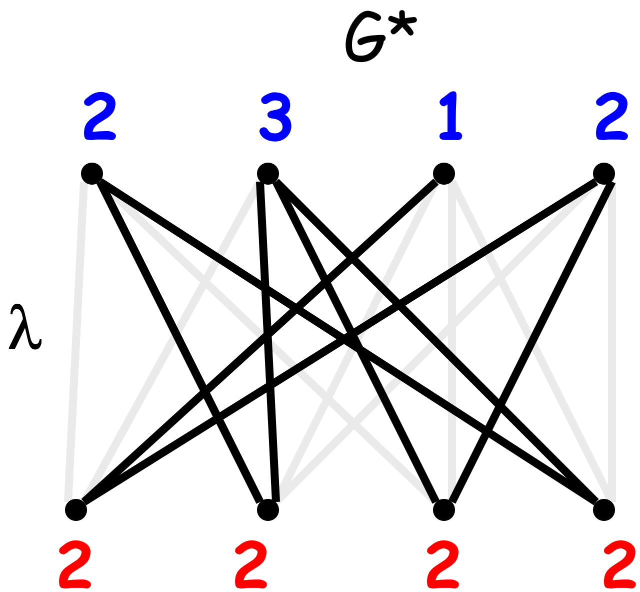
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: There exists a (u,v)-near-table similar to $G^*$.

# Simulated Annealing for BCT ?




Recall that

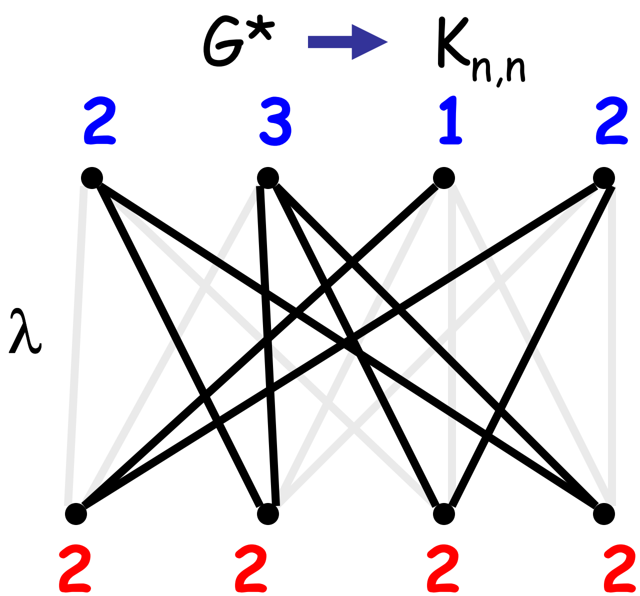
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathcal{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

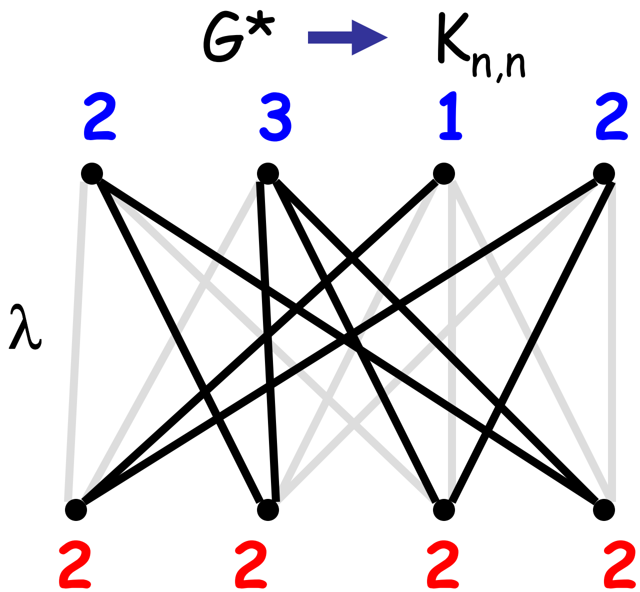
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathcal{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

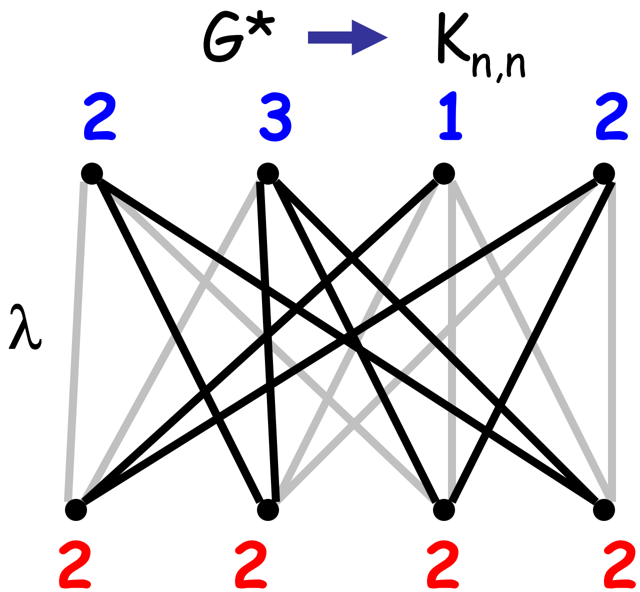
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

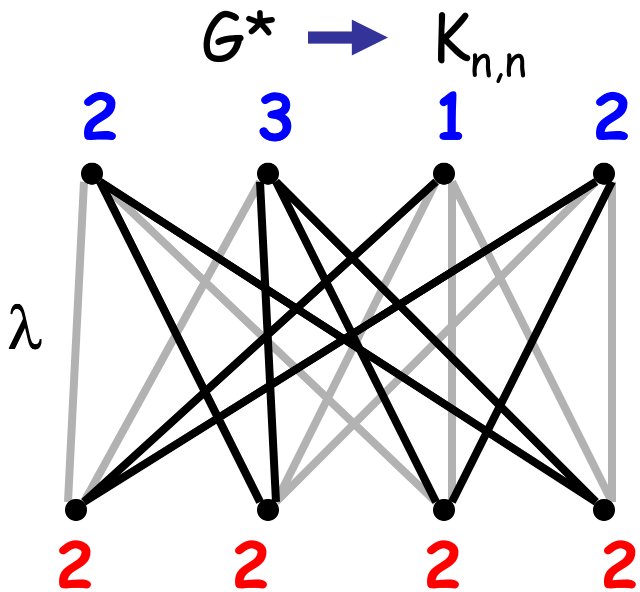
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

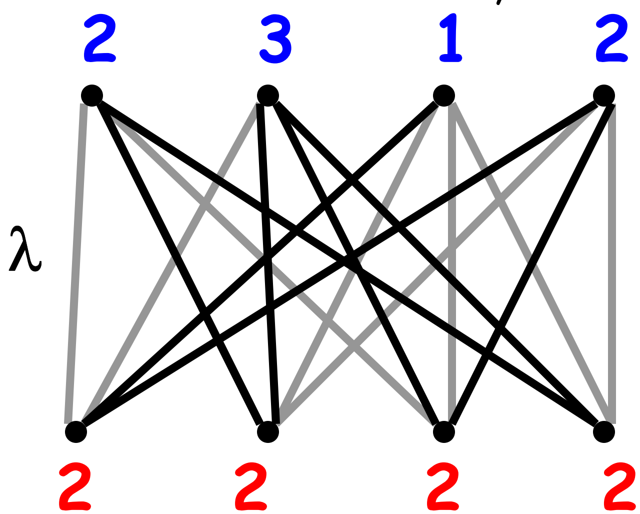
**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph G* with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?

$G^* \rightarrow K_{n,n}$




Recall that

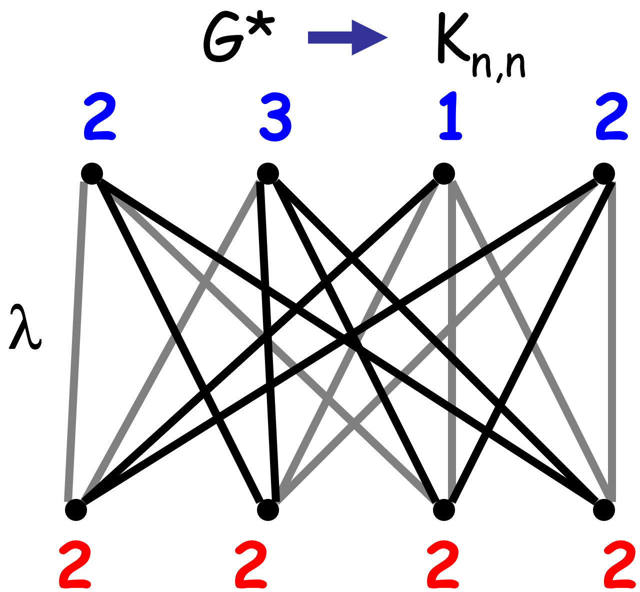
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

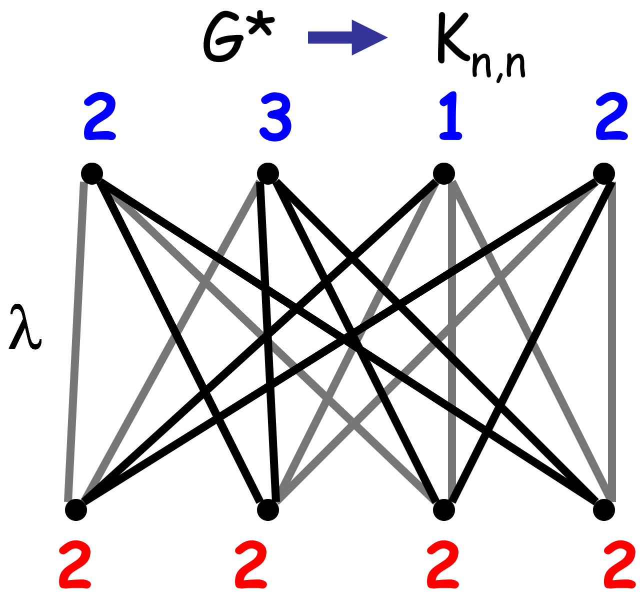
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph G* with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda$ = ~0.

# Simulated Annealing for BCT ?




Recall that

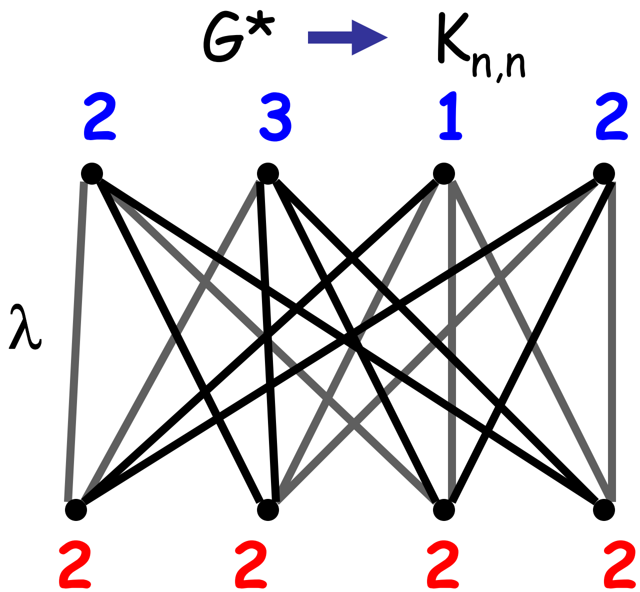
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph G* with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathcal{N}(u,v))$ "easy" to compute for $\lambda$ = ~0.

# Simulated Annealing for BCT ?




Recall that

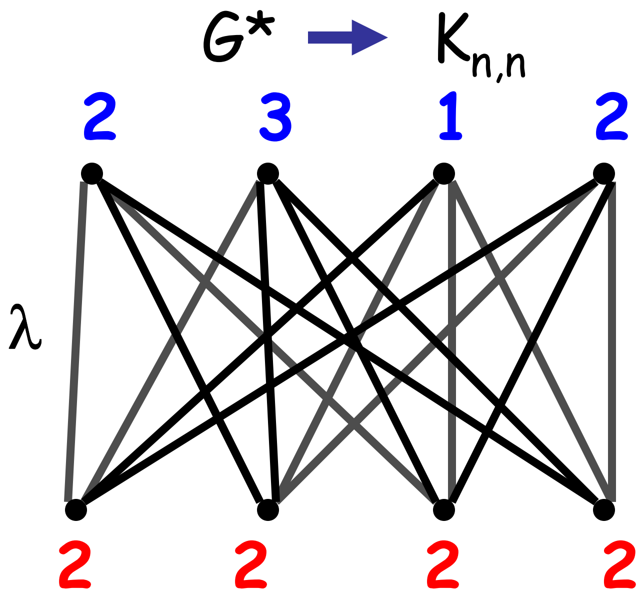
$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda$ = ~0.

# Simulated Annealing for BCT ?




Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

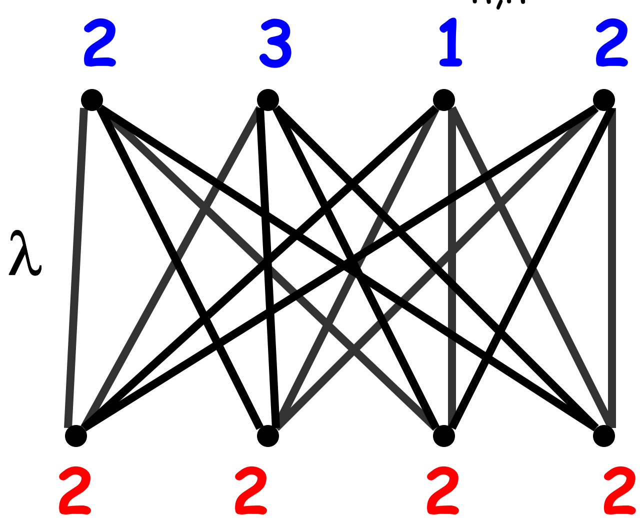
**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph G* with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?

$G^* \rightarrow K_{n,n}$




Recall that

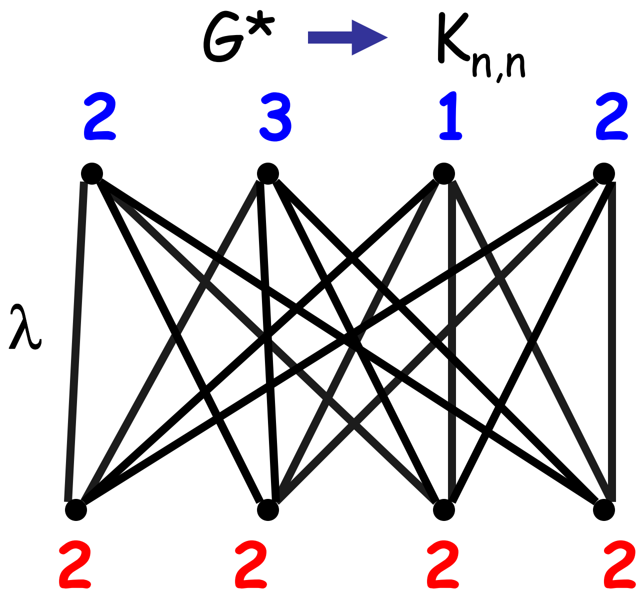
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathcal{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?




Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

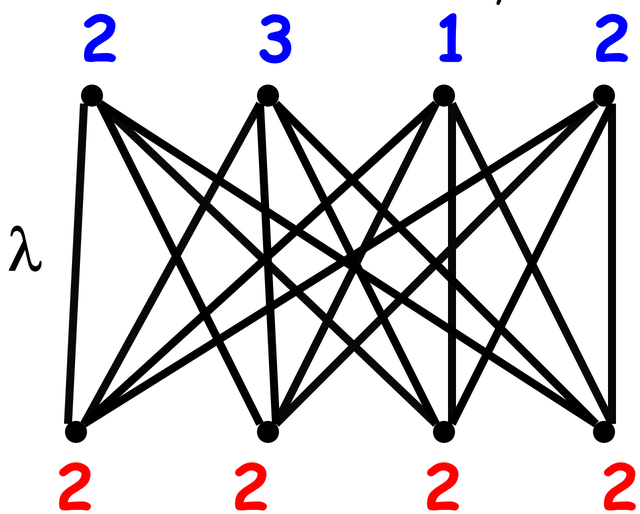
**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathscr{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?

$G^* \rightarrow K_{n,n}$




Recall that

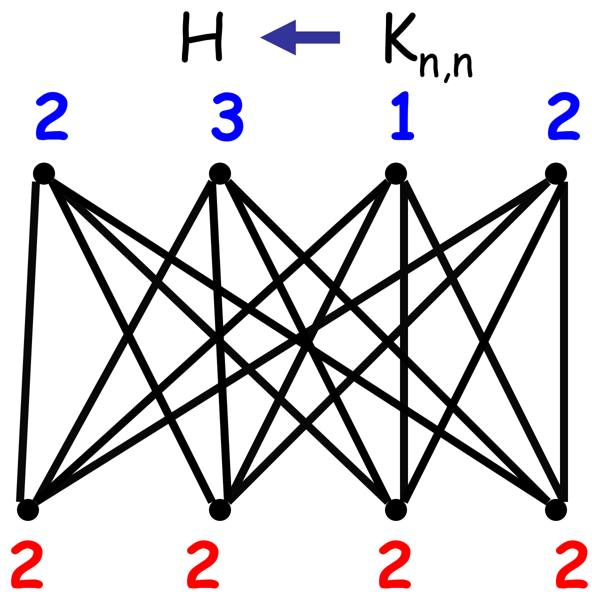
$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm** [Bezáková-Bhatnagar-Vigoda '06]:

There exists a graph $G^*$ with given degree sequence s.t. between any two vertices there exists an "alternating" path of length $\leq$ **5**.

**Corollary**: $\lambda(\mathcal{N}(u,v))$ "easy" to compute for $\lambda = \sim 0$.

# Simulated Annealing for BCT ?



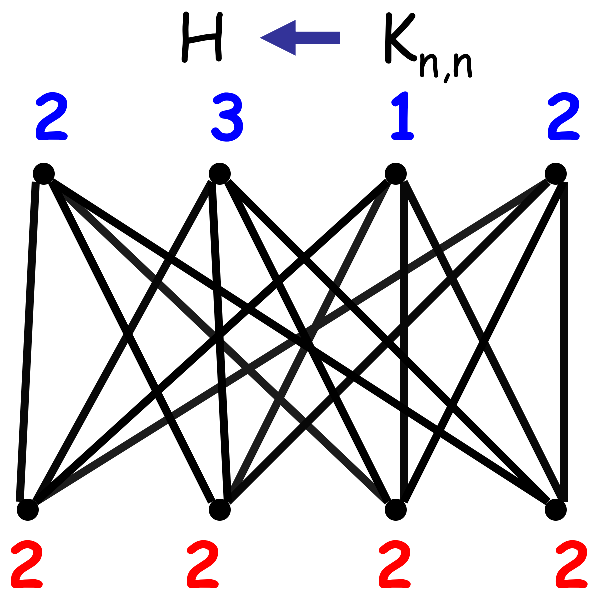

Recall that

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



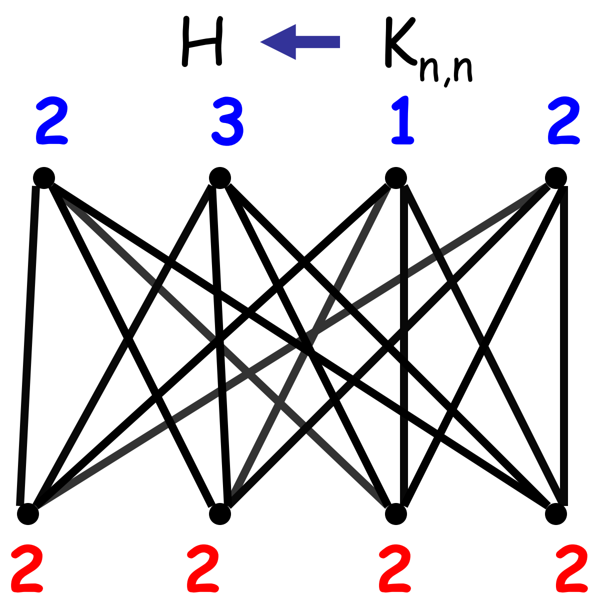

Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



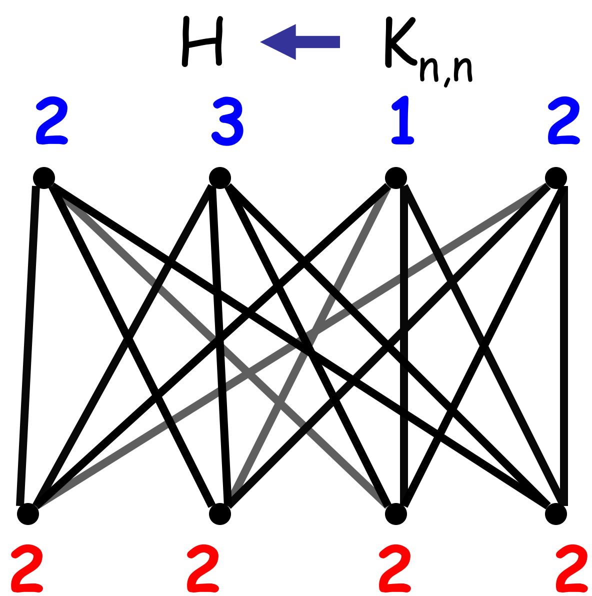

Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



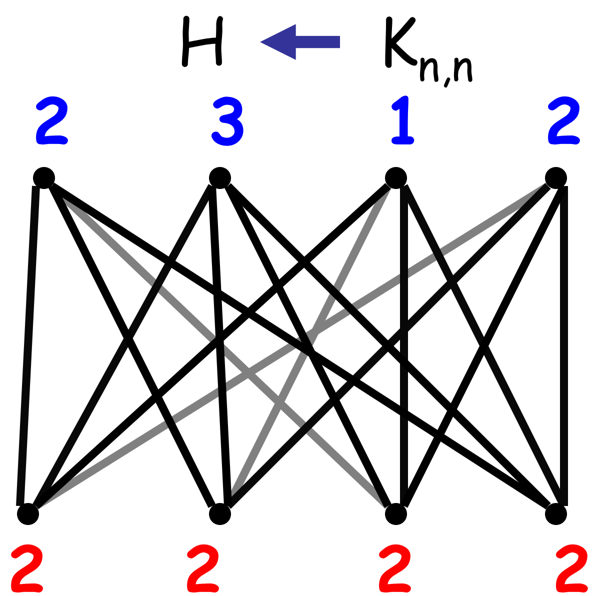

Recall that

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



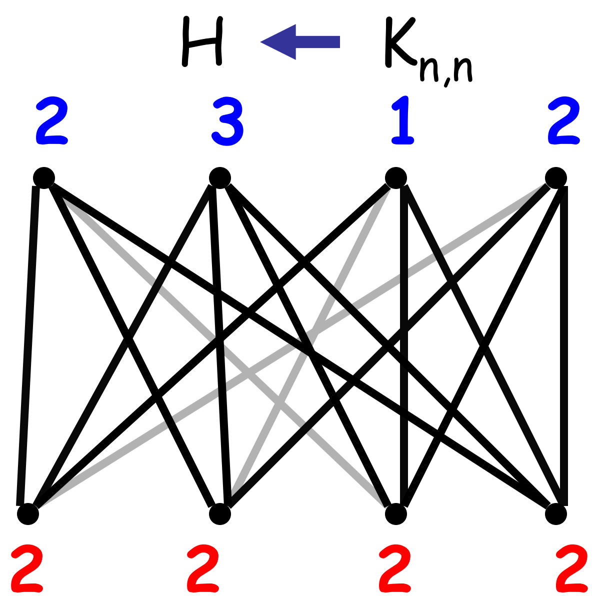

Recall that

$$\boxed{w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



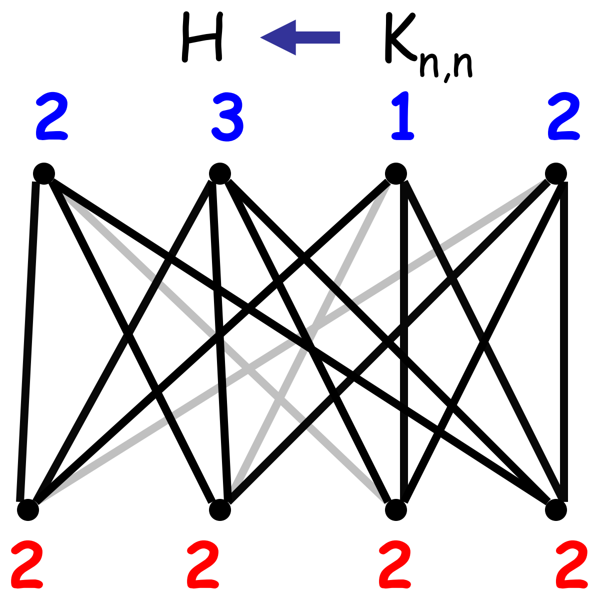

Recall that

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



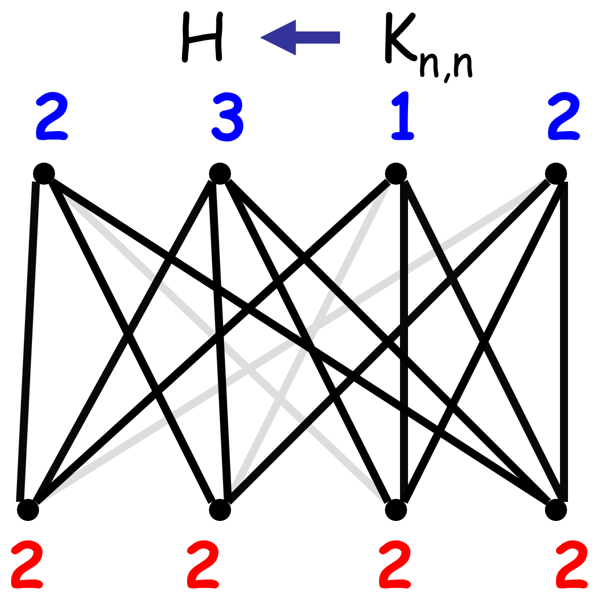

Recall that

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?



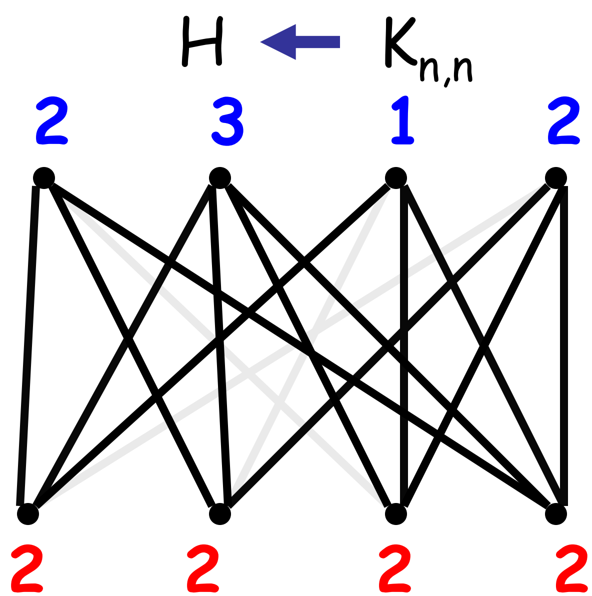

Recall that

$$w(u,v) = \frac{\lambda(\mathscr{P})}{\lambda(\mathscr{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Simulated Annealing for BCT ?




Recall that

$$w(u,v) = \frac{\lambda(\mathcal{P})}{\lambda(\mathcal{N}(u,v))}$$

**Thm:**

It is possible to sample/count bipartite graphs of given degree sequence (which are subgraphs of a given graph H) in time $O^*((nm)^3(n+m)^5)$.

# Different Approaches

**Theory** (Markov chain Monte Carlo with simulated annealing)

- Jerrum-Sinclair-Vigoda '01: approximate permanent in $O^*(n^{10})$, yields $O^*((mn)^{10})$ algorithm for m x n binary contingency tables
- Bezáková-Bhatnagar-Vigoda '06: $O^*((mn)^3(m+n)^5)$

**Practice** (sequential importance sampling, Chen-Diaconis-Holmes-Liu '05)

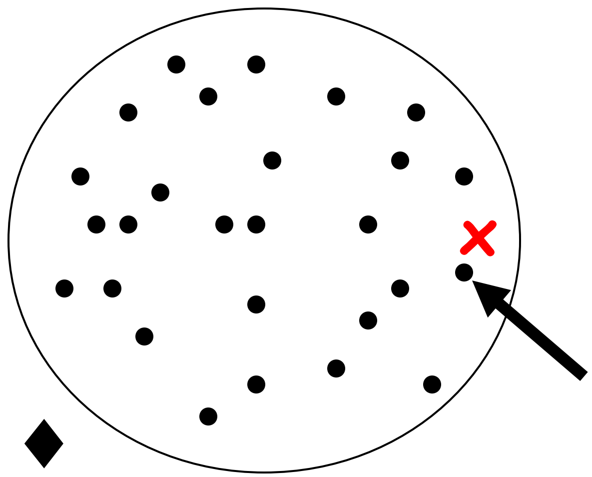
- Bezáková-Sinclair-Štefankovič-Vigoda '06: negative example
- Jose Blanchet '06: SIS works if marginals $O(n^{1/4})$
- Bayati-Kim-Saberi '07: alternative importance sampling method, works if marginals $O(n^{1/4})$

**Practice** (the switching Markov chain, Diaconis-Gangolli '94)

- Kannan-Tetali-Vempala '97, Cooper-Dyer-Greenhill '05: works for regular marginals

# Importance Sampling

for counting problems



Probability distribution $\sigma$
on the points + ♦

with positive probability $\sigma(x)>0$

Random variable

$$\eta(s) = \begin{cases} 1/\sigma(s) & \text{if } s \text{ in the set} \\ 0 & \text{if } s \text{ is } \blacklozenge \end{cases}$$

Unbiased estimator

$$E[\eta] = \sum \sigma(x).1/\sigma(x) = \text{size of the set}$$

# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | 4 |
| | | | | | | | 2 |
| | | | | | | | 3 |
| | | | | | | | 5 |
| | | | | | | | 3 |
| 3 | 4 | 2 | 1 | 2 | 2 | 3 | |

# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

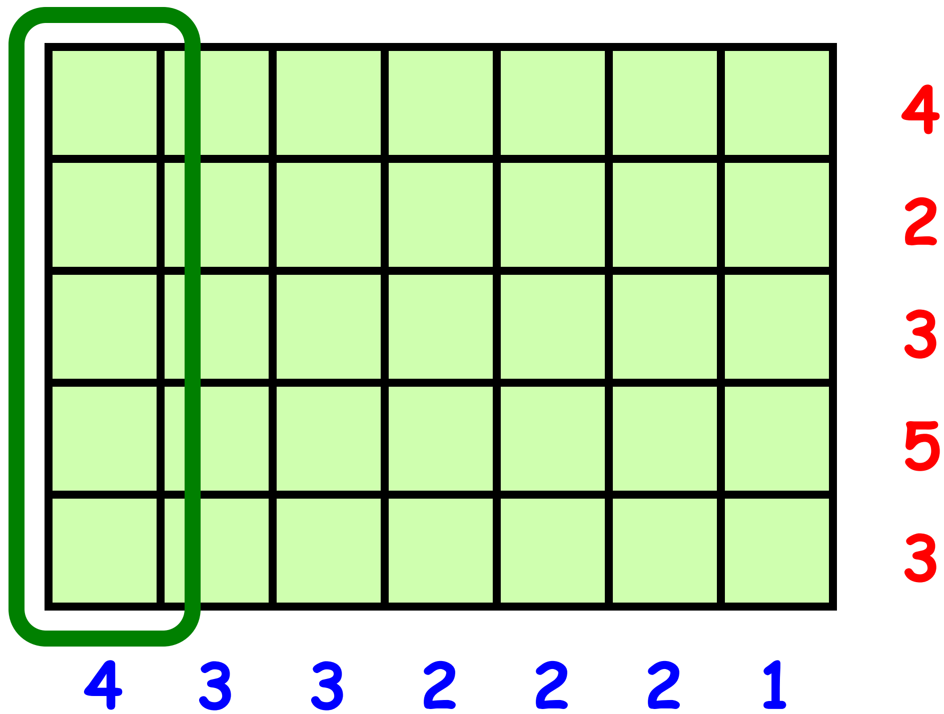
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | 4 |
| | | | | | | | 2 |
| | | | | | | | 3 |
| | | | | | | | 5 |
| | | | | | | | 3 |
| 4 | 3 | 3 | 2 | 2 | 2 | 1 | |

# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific σ

- fill table column-by-column
- assign each column ignoring other column sums

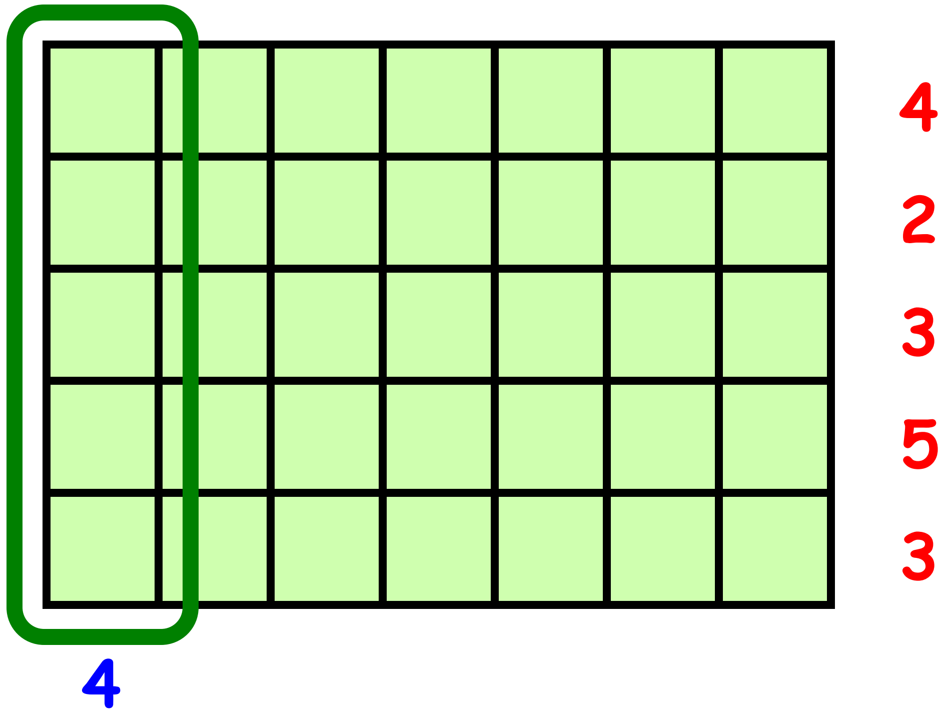


# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

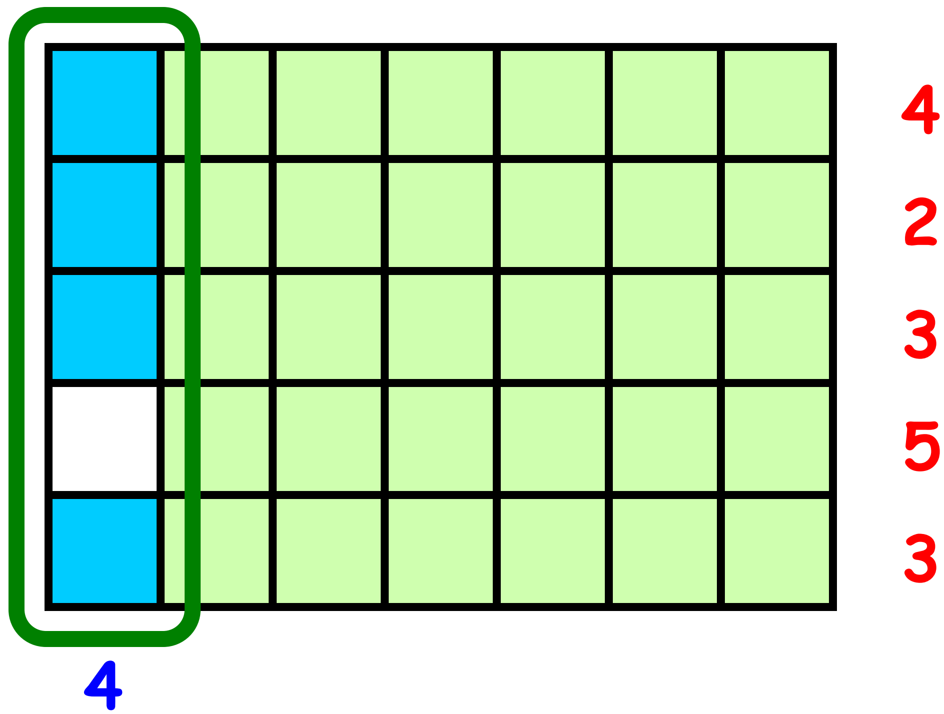
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

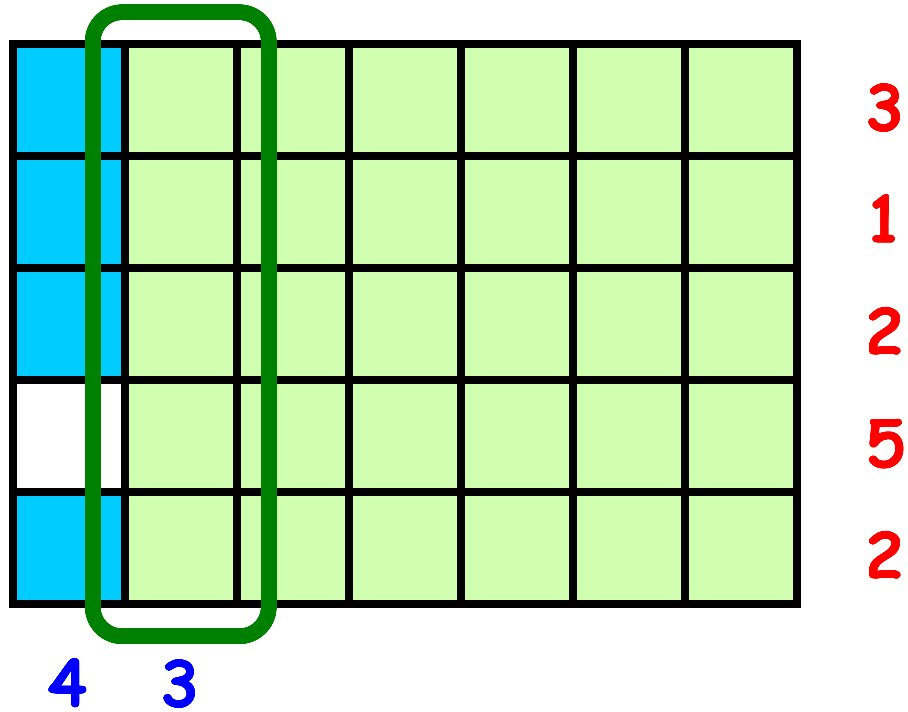
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

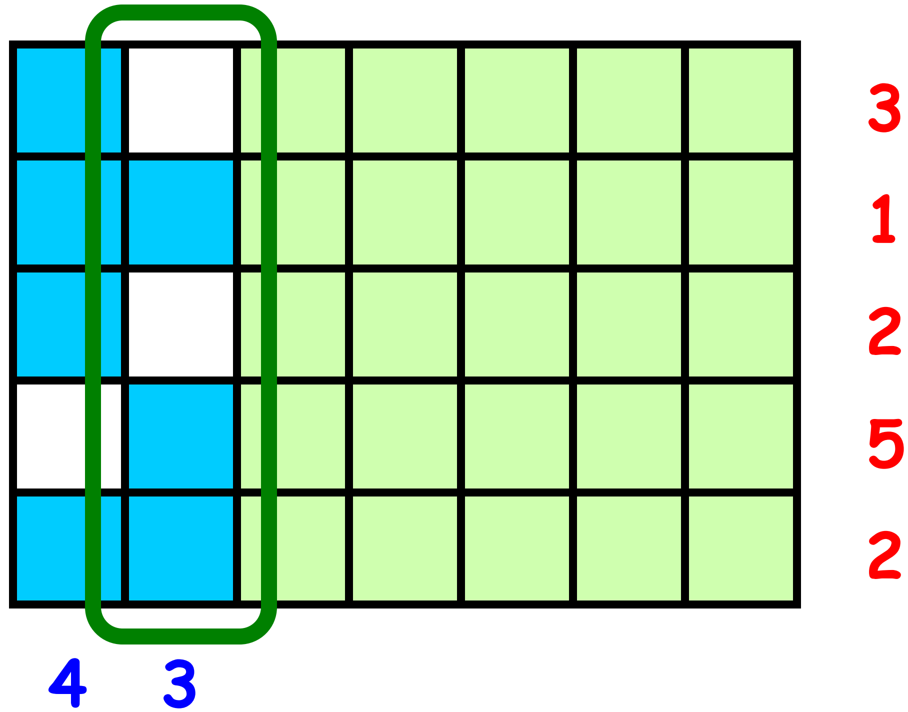
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

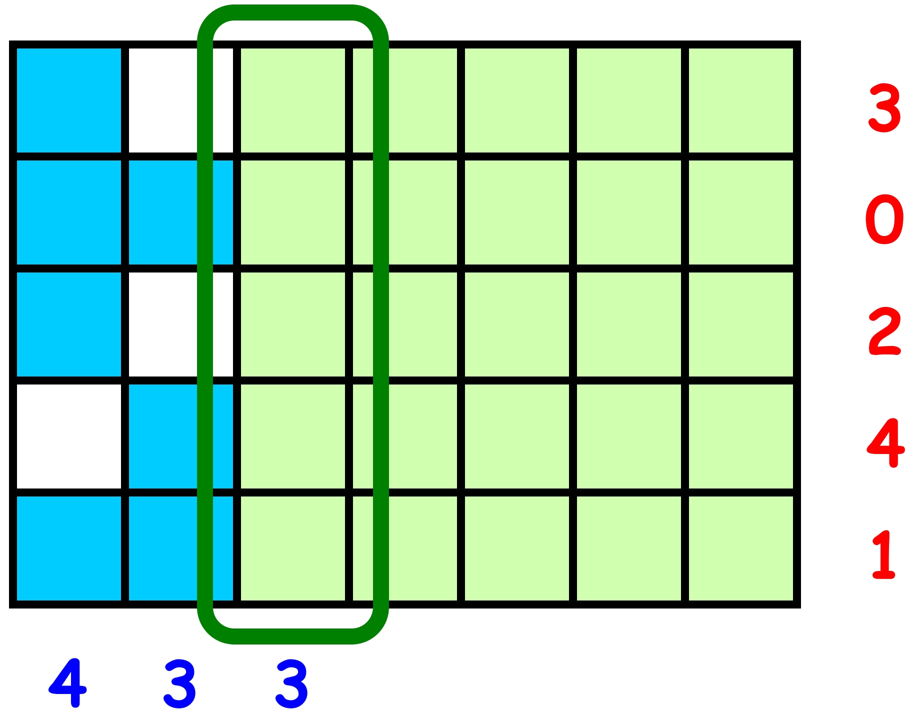
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

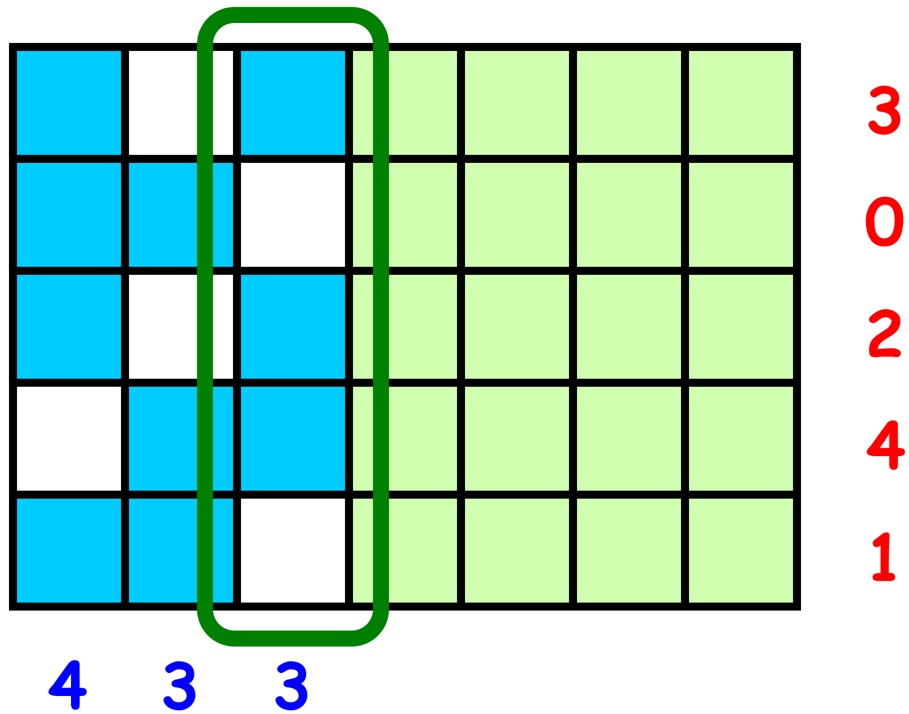
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

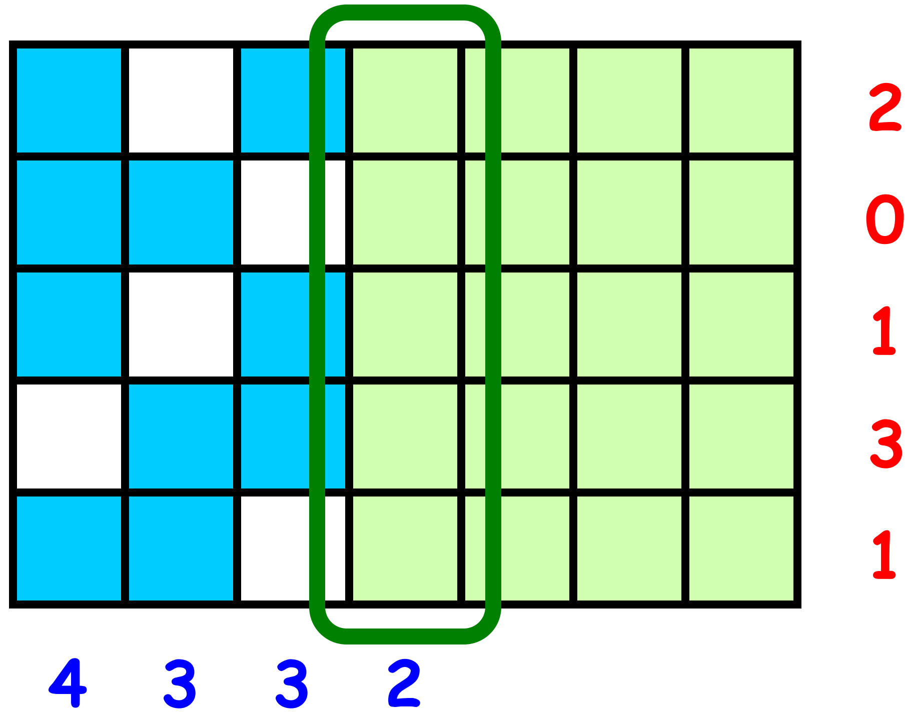
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

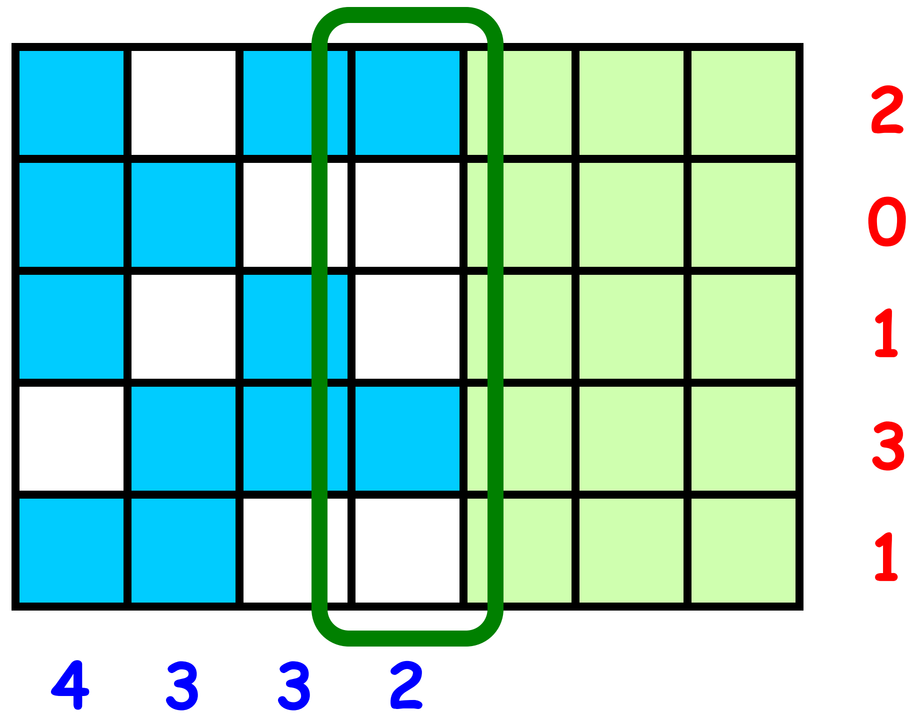
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column
with probability
proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product
ranges over **i: rows
with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

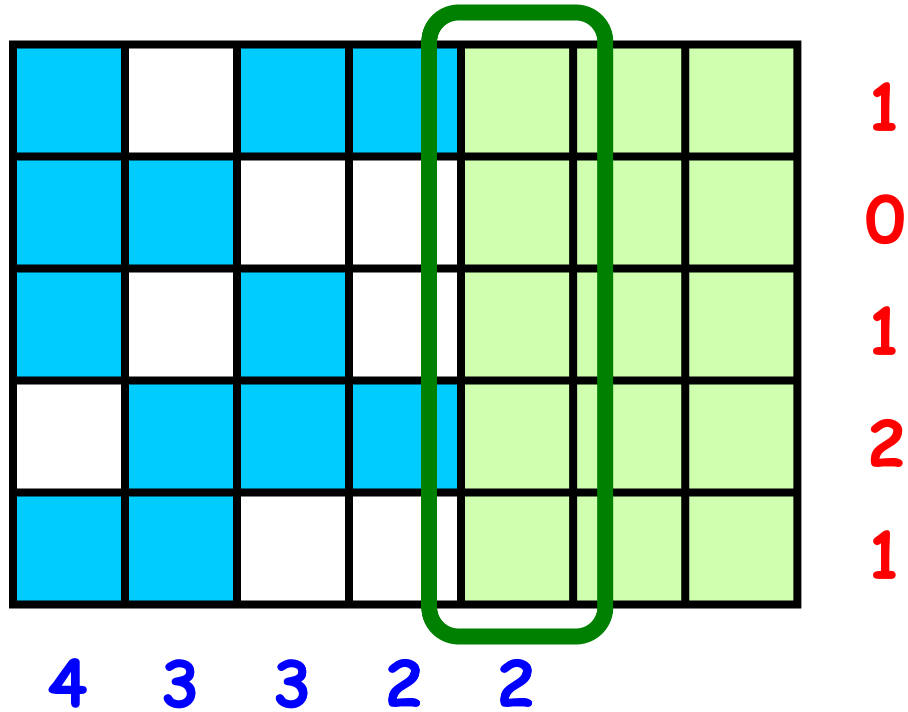
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

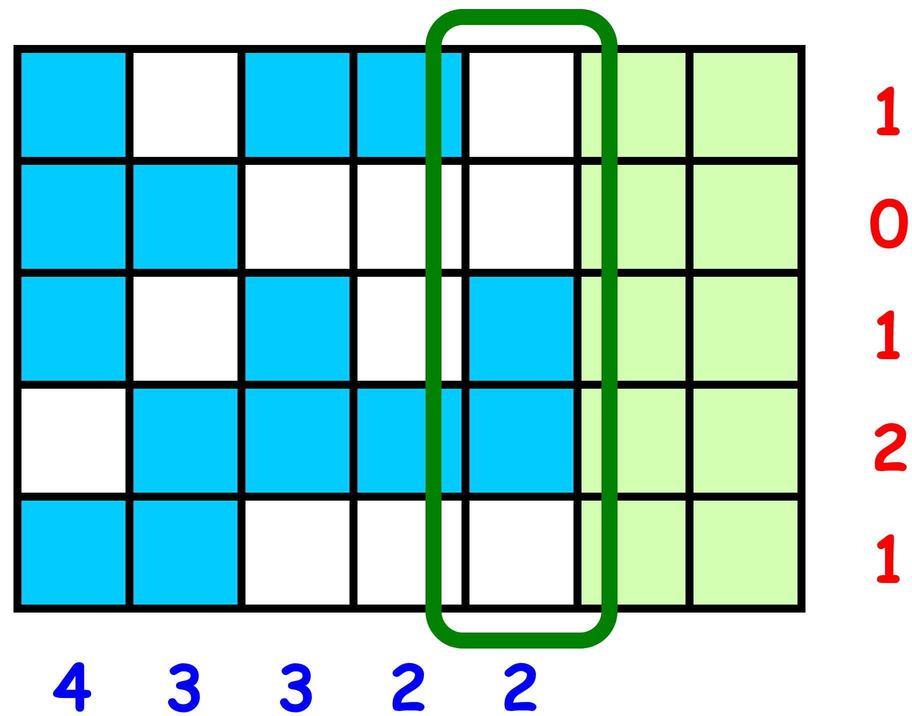
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

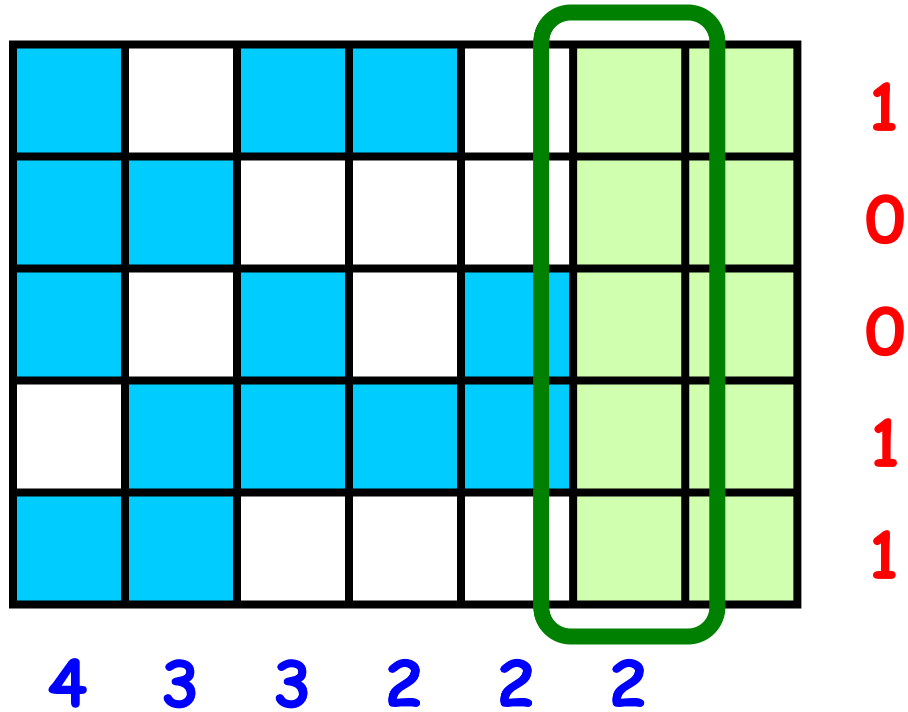
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\prod r_i/(n-r_i)$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

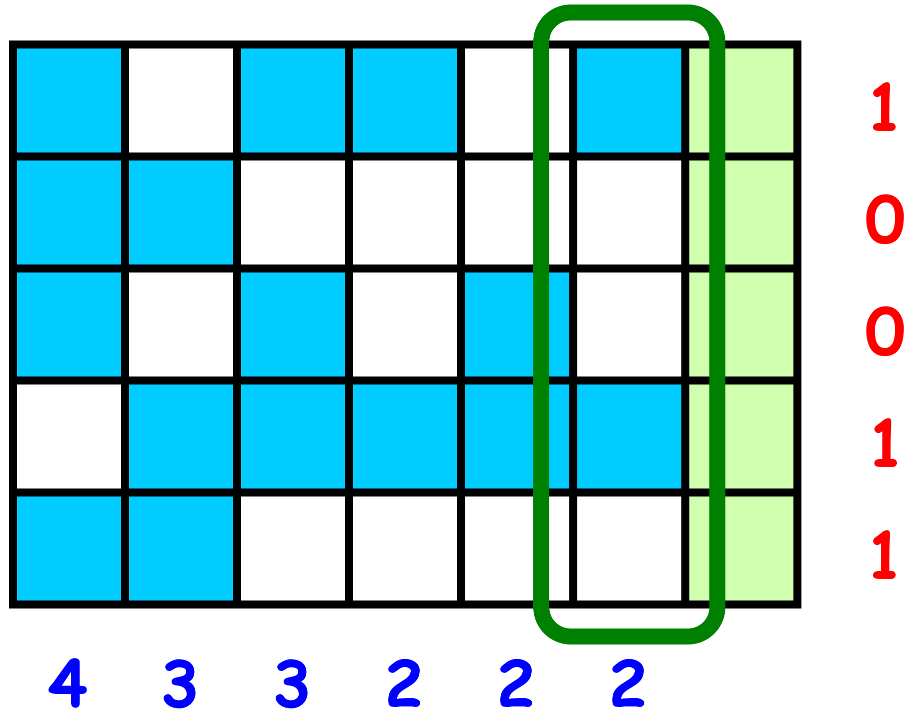
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

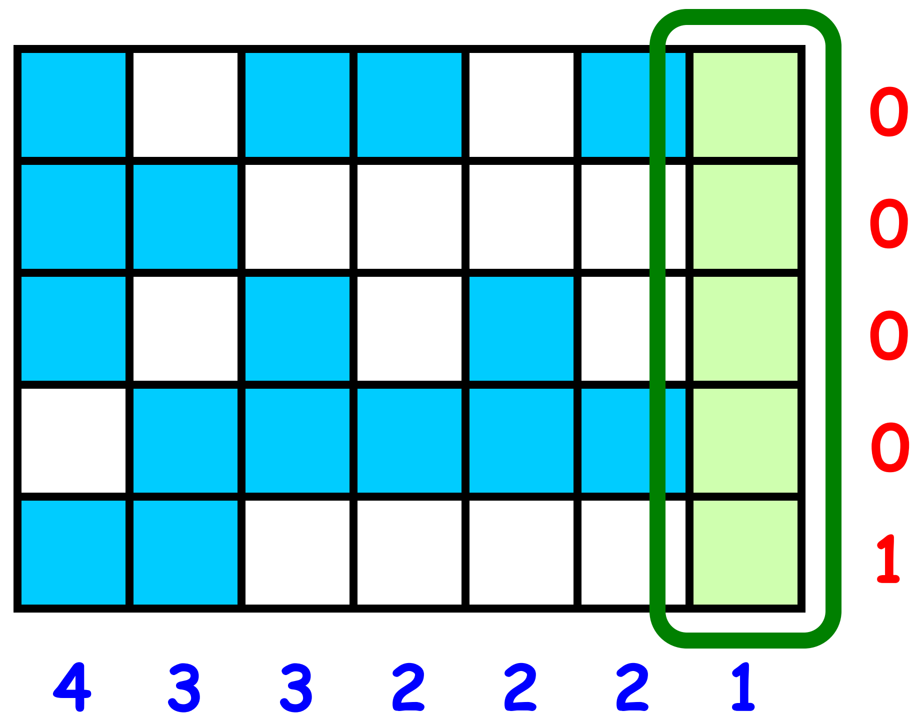
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\prod r_i/(n-r_i)$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

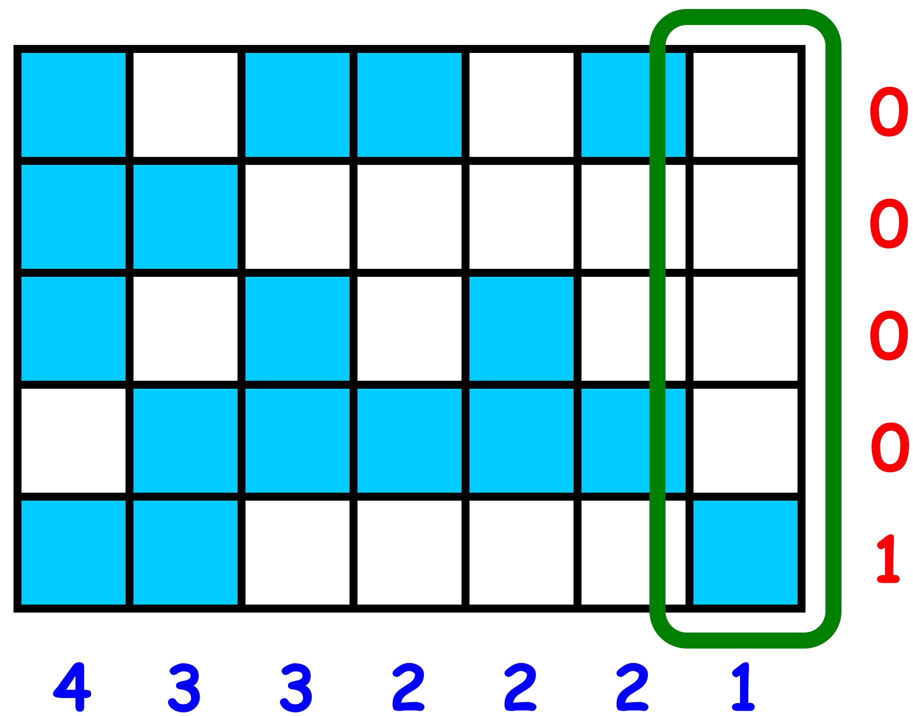
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

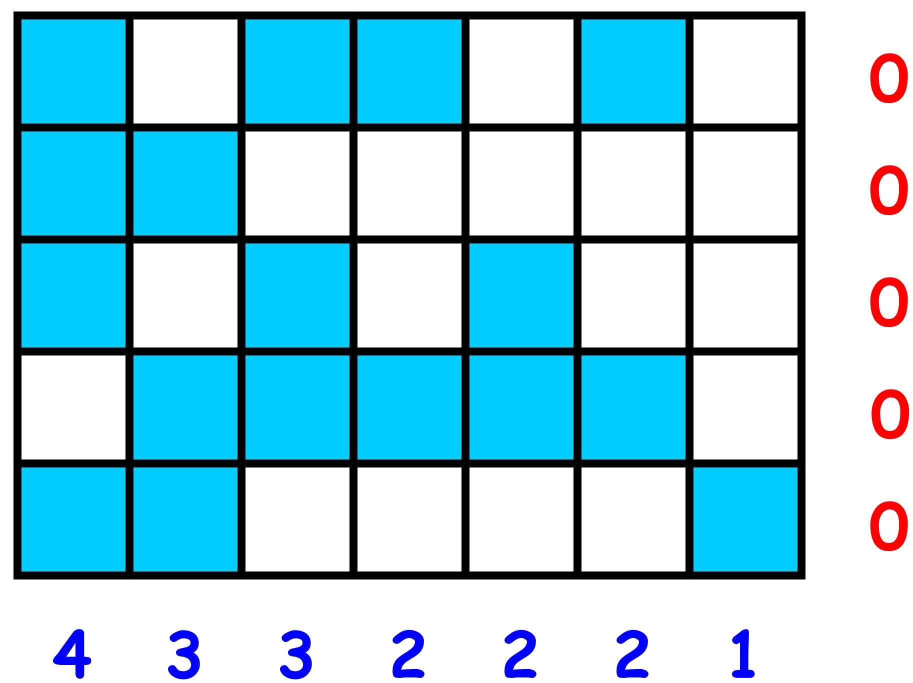
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

where product ranges over **i: rows with assignment 1**



# Sequential Importance Sampling for BCT

[Chen-Diaconis-Holmes-Liu '05]

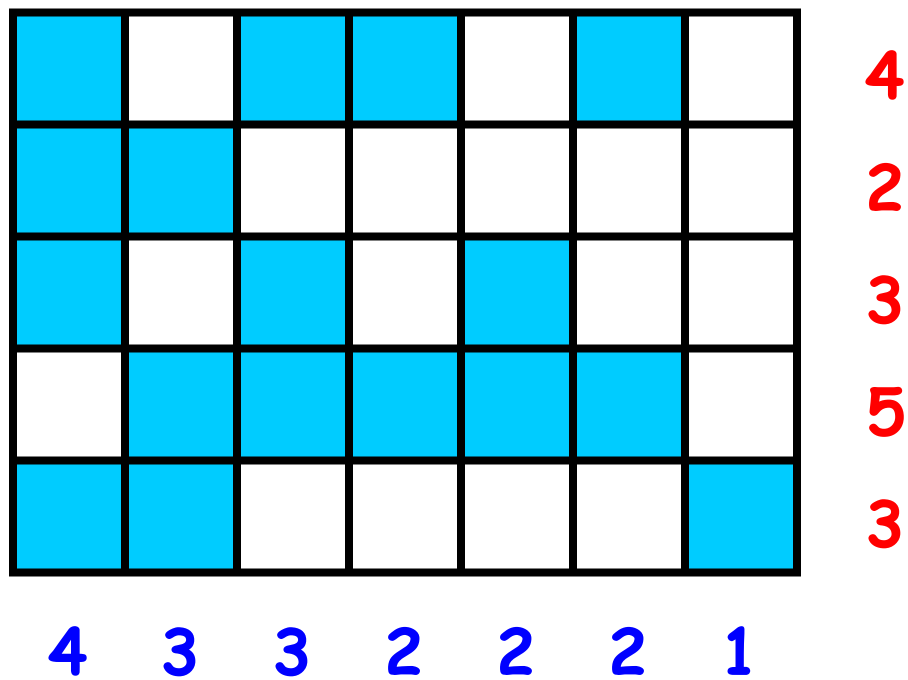
a specific $\sigma$

- fill table column-by-column
- assign each column ignoring other column sums

assign the column with probability proportional to

$$\boxed{\prod r_i/(n-r_i)}$$

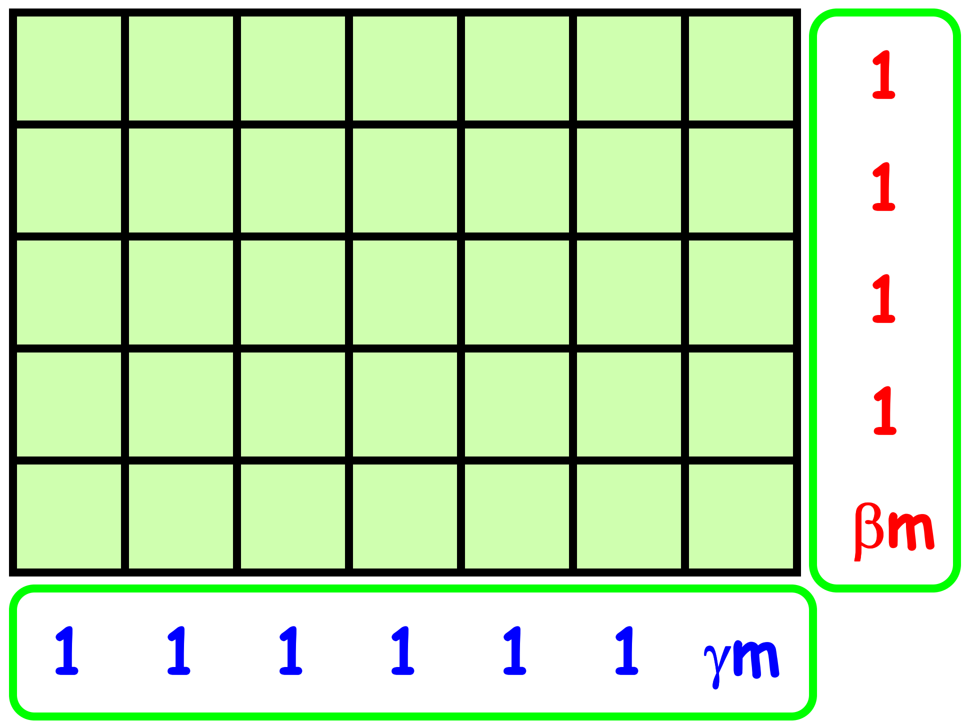
where product ranges over **i: rows with assignment 1**



# A Counterexample for SIS

**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

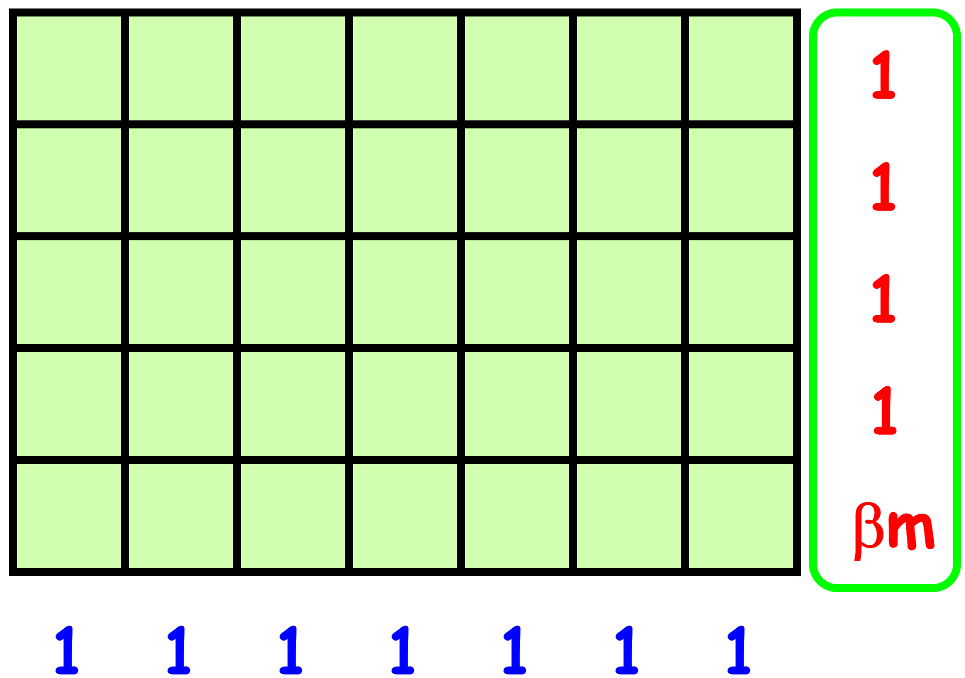


# A Counterexample for SIS

**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any β, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

## Simpler example



# A Counterexample for SIS

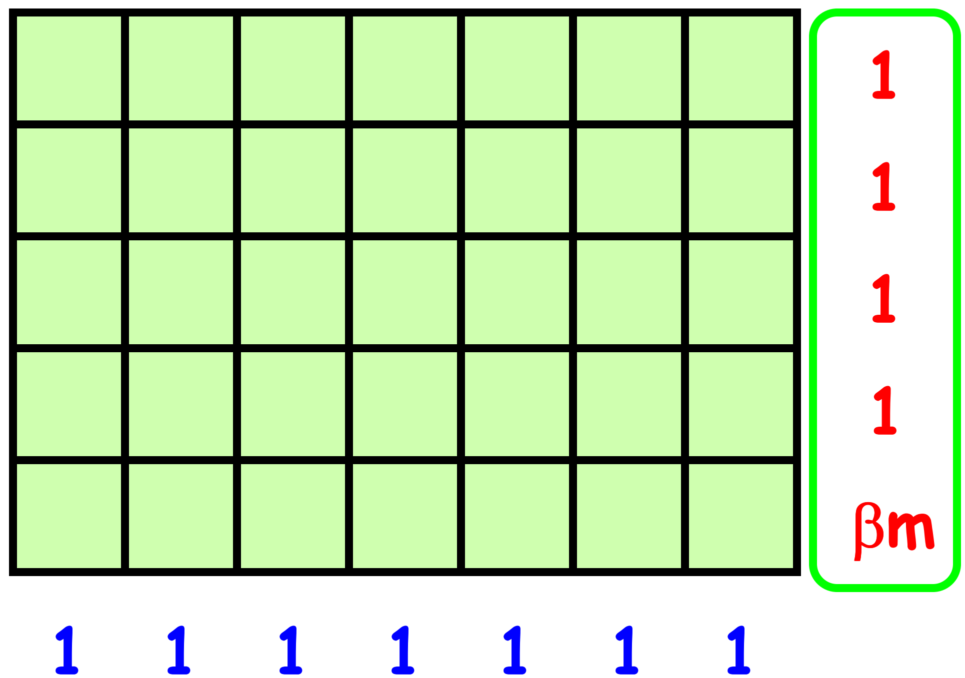
**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

## Intuition

Random table:

- randomly choose $\beta$**m** ones



# A Counterexample for SIS

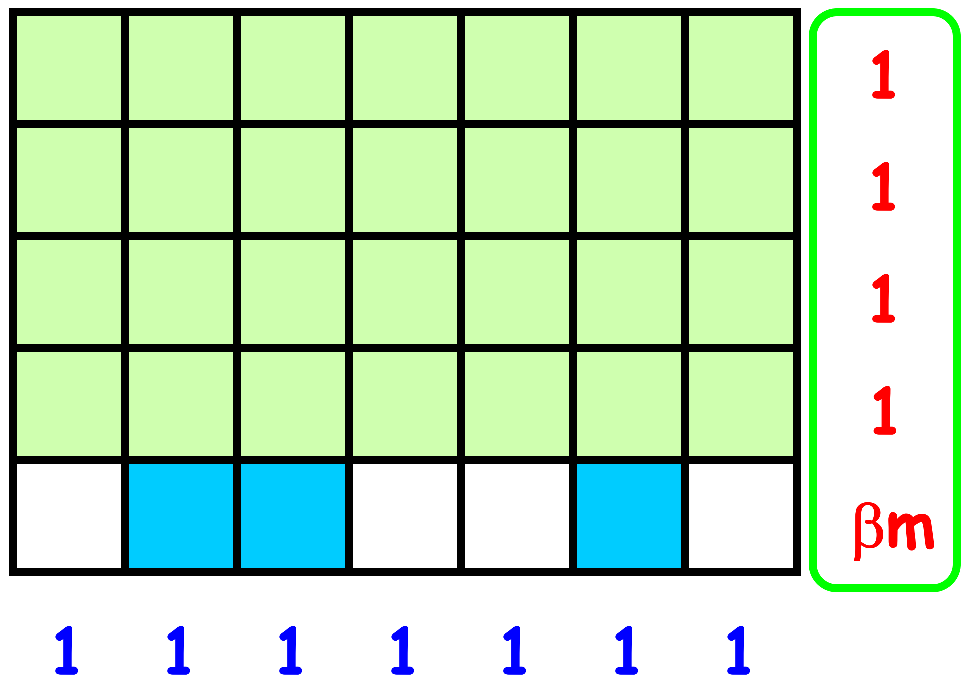
**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

## Intuition

Random table:

- randomly choose $\beta$**m** ones



# A Counterexample for SIS

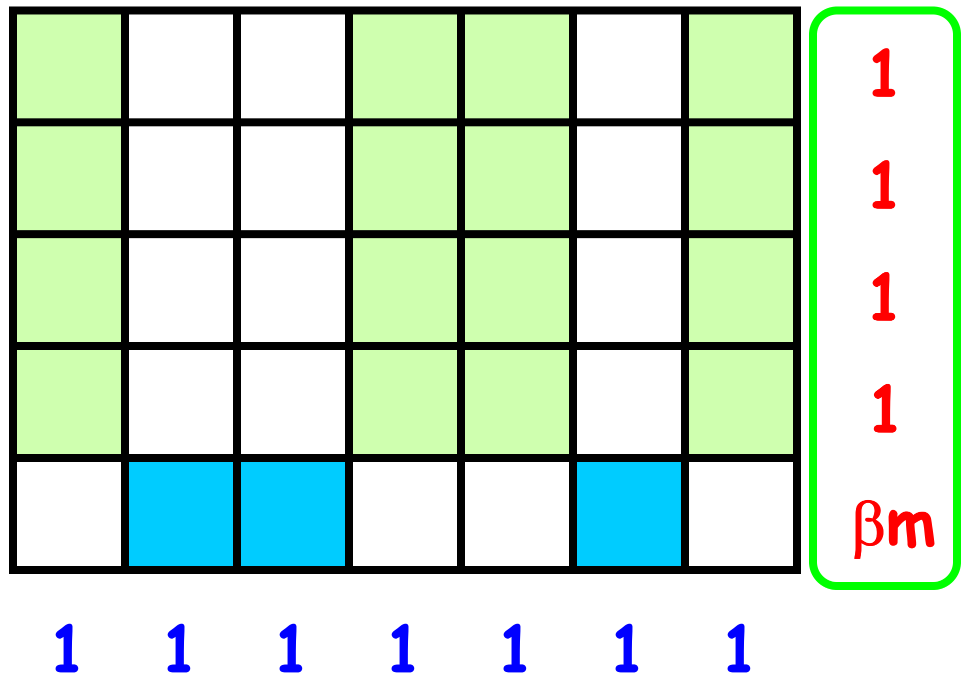
**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any β, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

## Intuition

Random table:

- randomly choose βm ones



# A Counterexample for SIS

**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any β, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

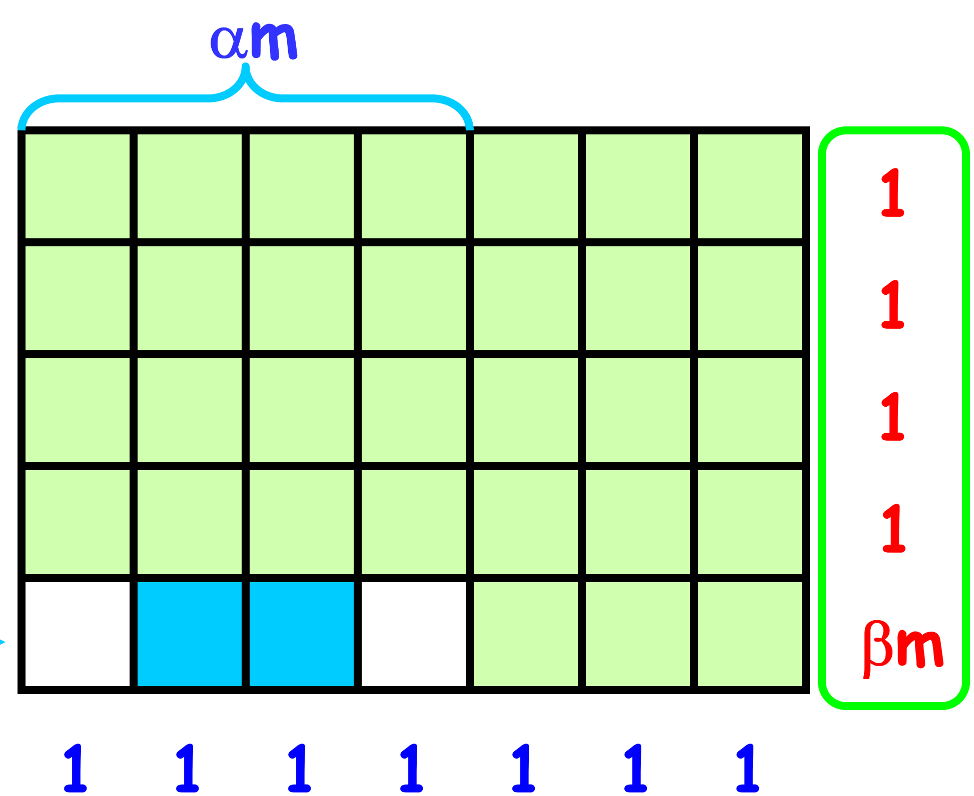
## Intuition

Expect: $\alpha\beta m$ ones

SIS: asymptotically fewer

Random table:

- randomly choose $\beta m$ ones



# A Counterexample for SIS

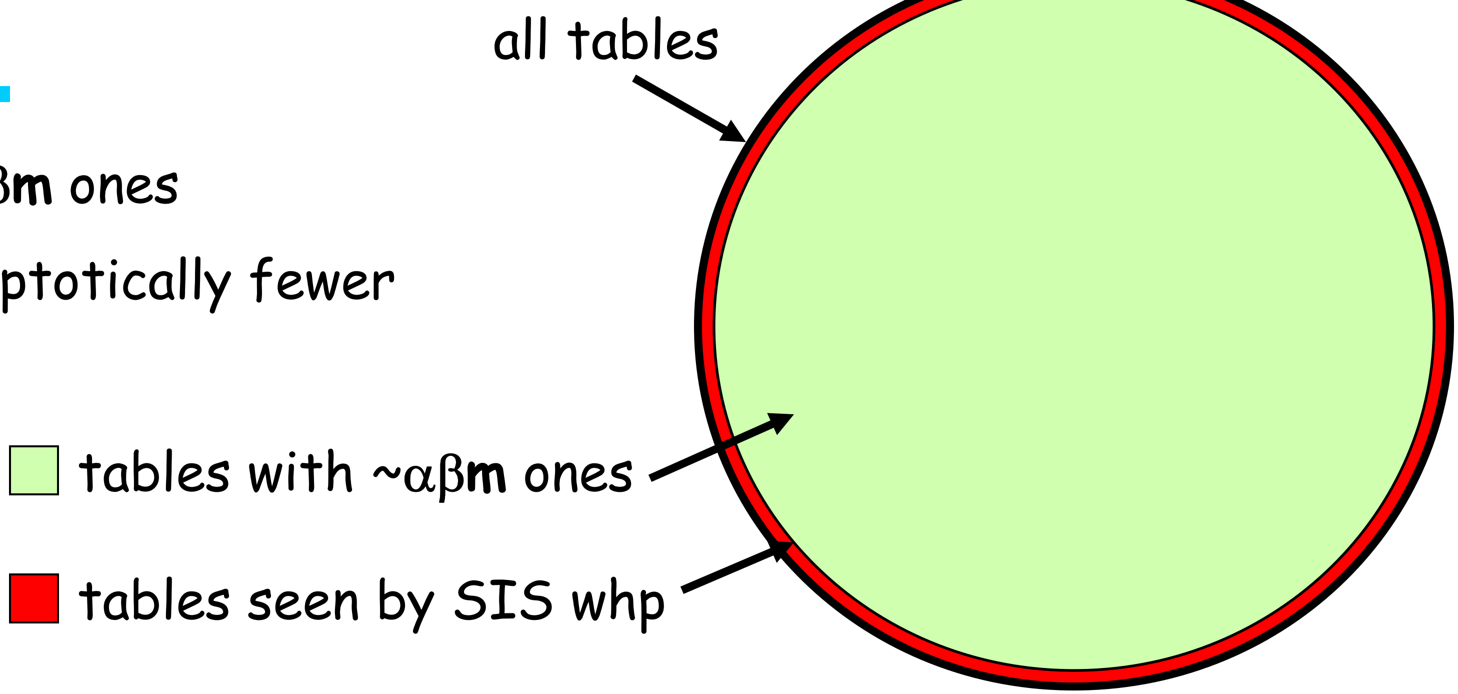
**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

For any $\beta$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

## Intuition

Expect: $\alpha\beta\mathbf{m}$ ones

SIS: asymptotically fewer



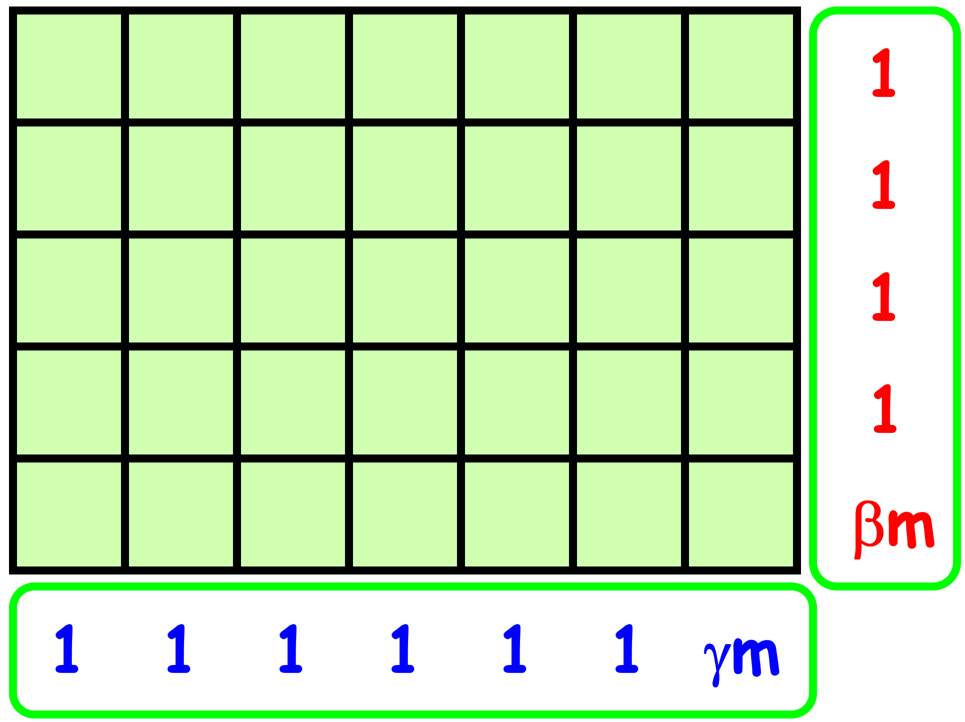
# A Counterexample for SIS

**Thm** [Bezáková-Sinclair-Štefankovič-Vigoda '06]:

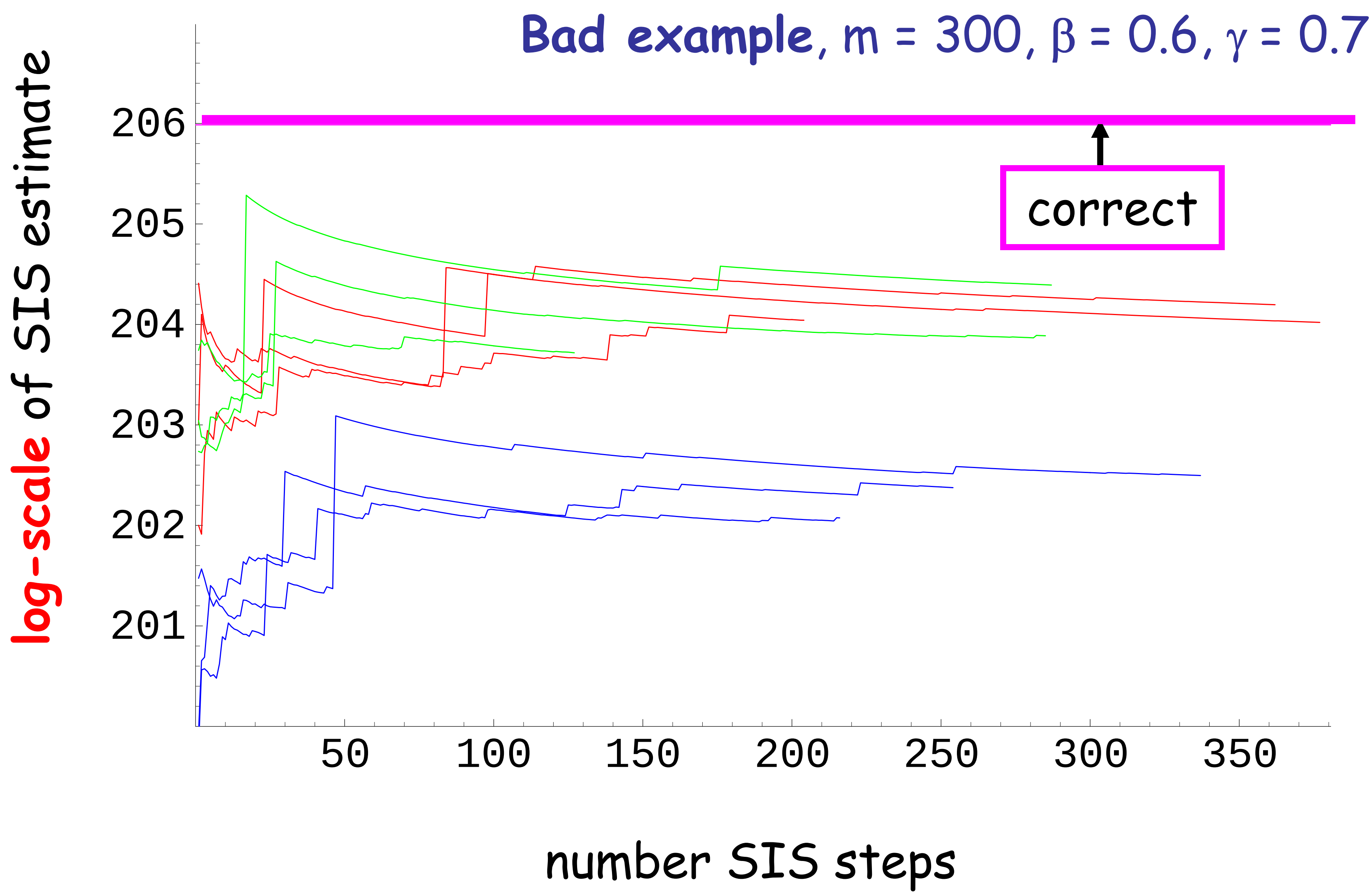
For any $\beta \neq \gamma$, SIS output after any subexponential number of trials is **off by an exponential factor** (with high probability).

Result holds for any order of rows/columns.

Alternating rows and columns?



# SIS - Experimental Results



# Open Problems

- Practical algorithm ?
- Detecting convergence of SIS
- SIS for larger marginals ?
- The Switching Markov chain of Diaconis-Gangolli ?

- General contingency tables
- Cell-bounded tables
- Counting non-bipartite graphs with a given degree sequence