CFTC Centro de Física Teórica e Computacional

Ciências ULisboa

Percolation theory

Nuno Araújo

Departamento de Física, Faculdade de Ciências, Universidade de Lisboa, Portugal

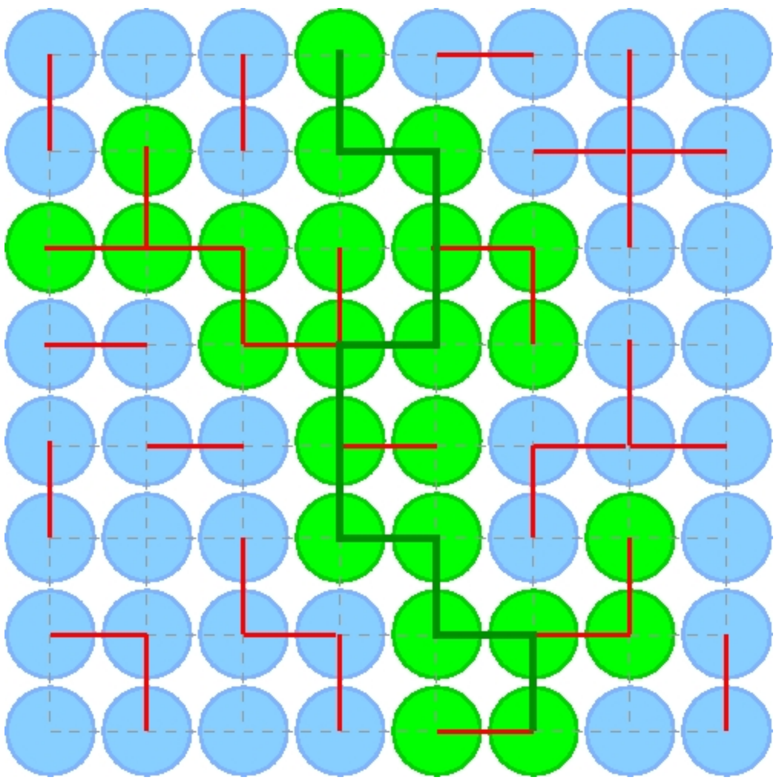
Centro de Física Teórica e Computacional, Universidade de Lisboa, Portugal

http://www.namaraujo.net

Percolation model

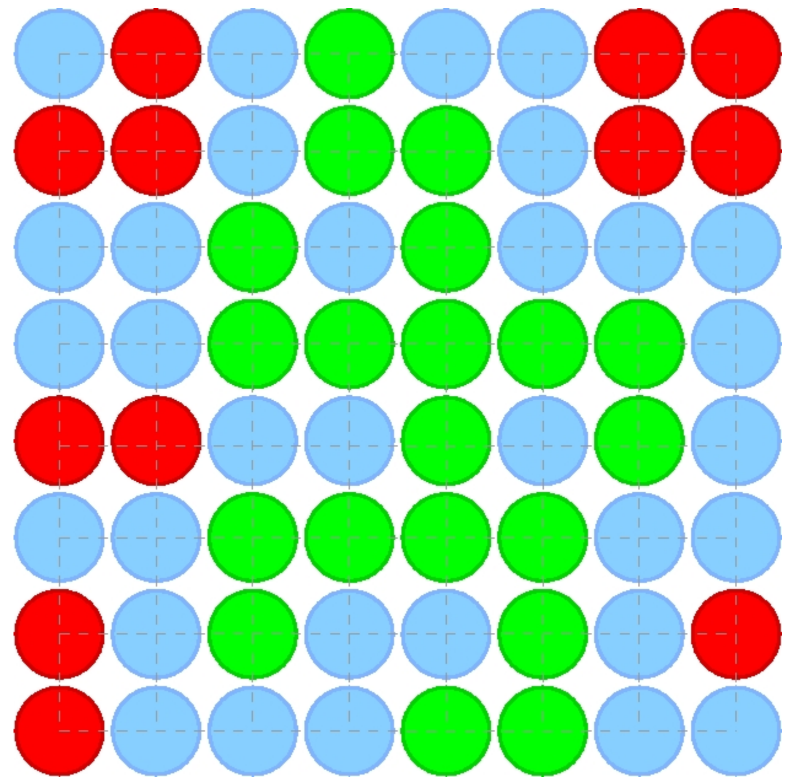
$$p^{O}(1-p)^{E}$$

Bonds



$$2^{N_{Bonds}}$$

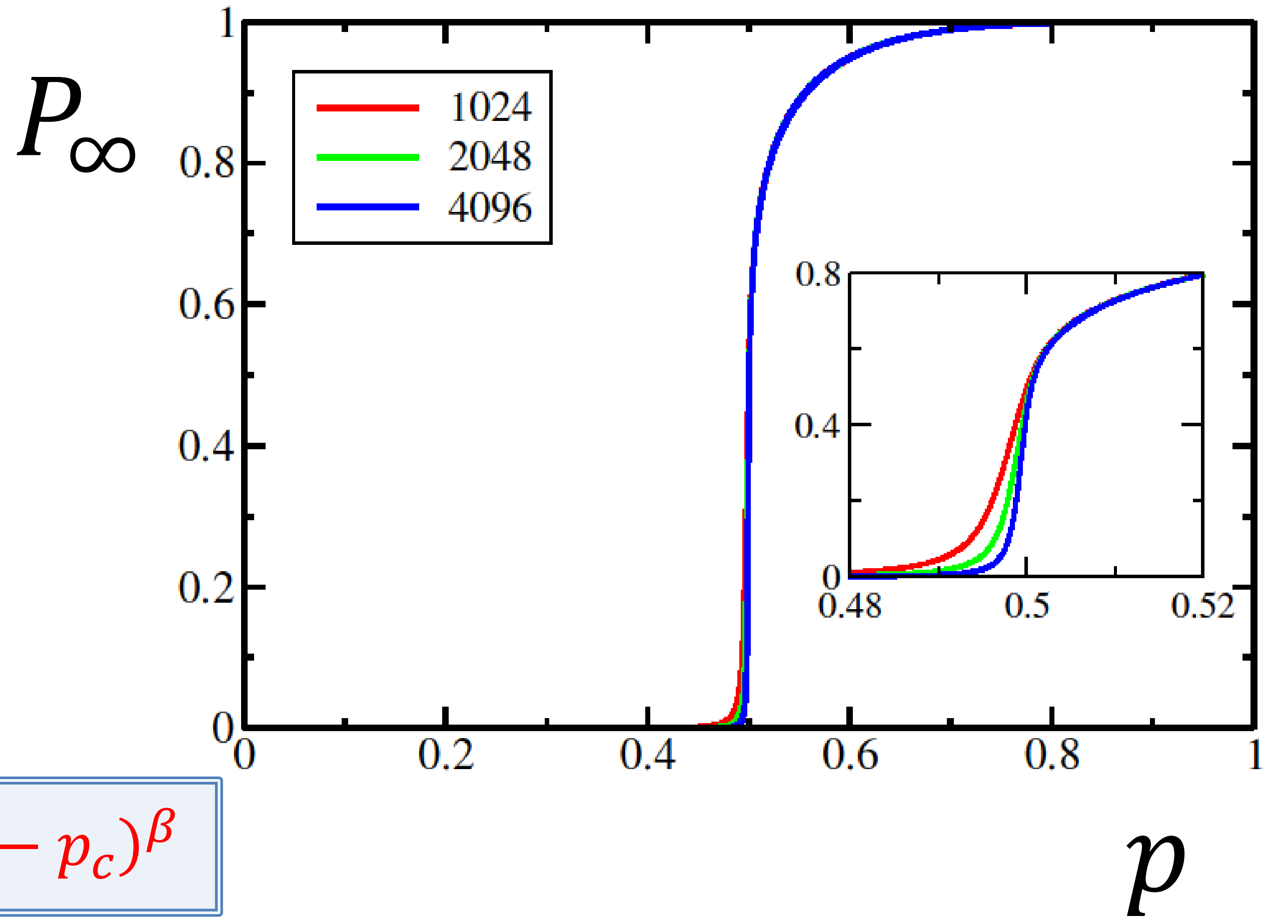
Sites



$$2^{N_{Sites}}$$

Percolation model
order parameter

$$P_\infty = \frac{s_{max}}{N}$$

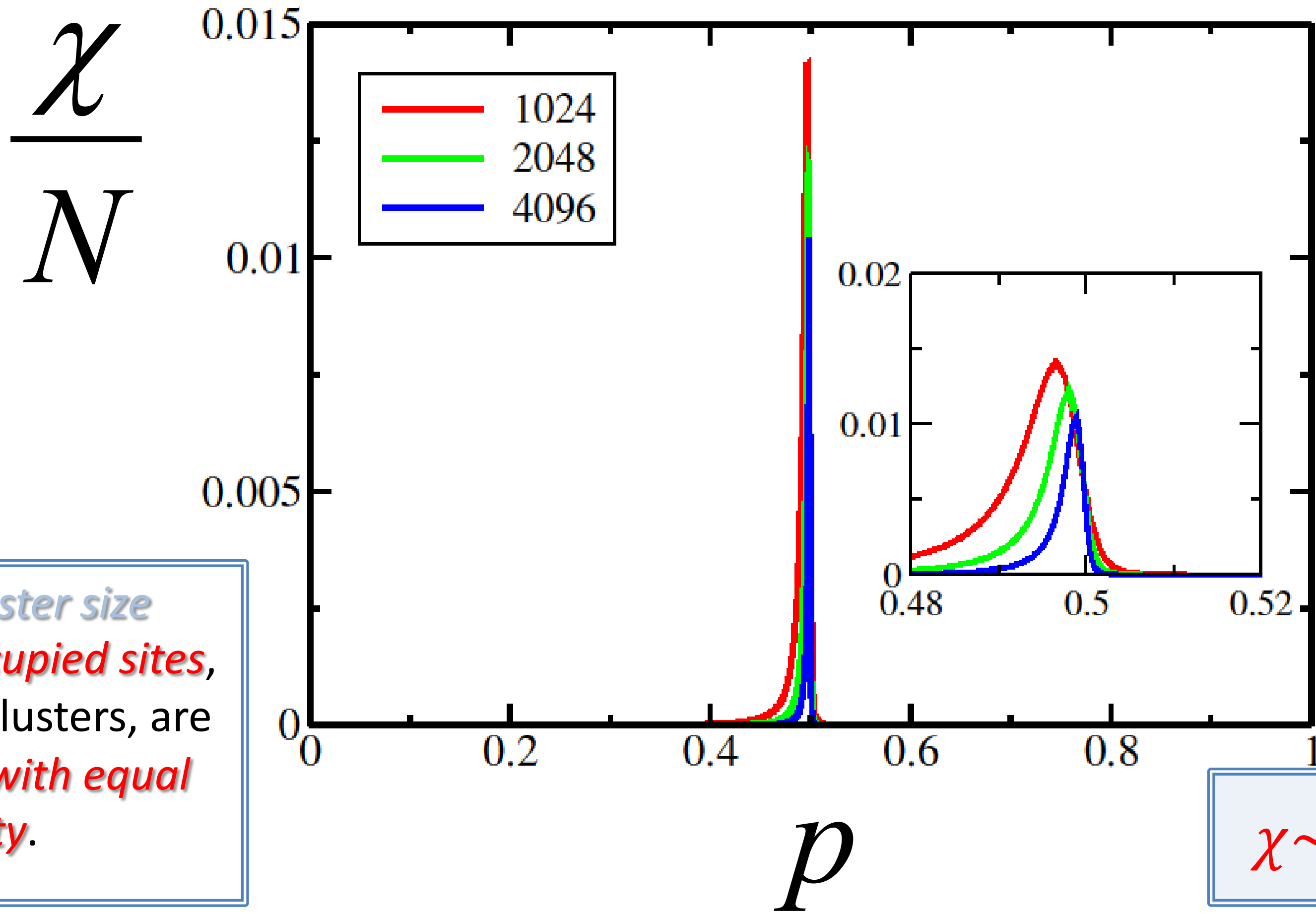


$P_\infty \sim (p - p_c)^\beta$

Percolation model

$$\chi = \frac{1}{N}\sum_{i\neq \max} s_i^2$$

fluctuations (mean cluster size)



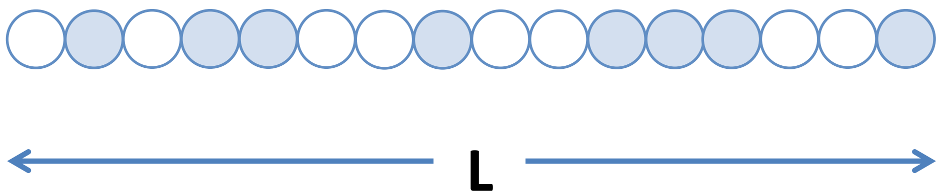
Mean cluster size when *occupied sites*, and not clusters, are *selected with equal probability*.

$$\chi \sim (p_c - p)^{-\gamma}$$

Exact solution in one dimension

cluster number density

p	occupied
1-p	empty

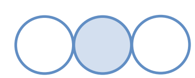


Cluster number frequency:

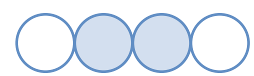
$$N(s,p;L) = L(1-p)^2 p^s$$

Probability that a site belongs to a cluster of size s :

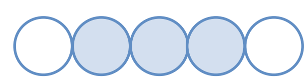
$s = 1$: $(1-p)p(1-p) = p(1-p)^2$



$s = 2$: $2(1-p)p^2(1-p) = 2p^2(1-p)^2$



$s = 3$: $3(1-p)p^3(1-p) = 3p^3(1-p)^2$



...

$s(1-p)p^s(1-p) = sp^s(1-p)^2$

Cluster number density:

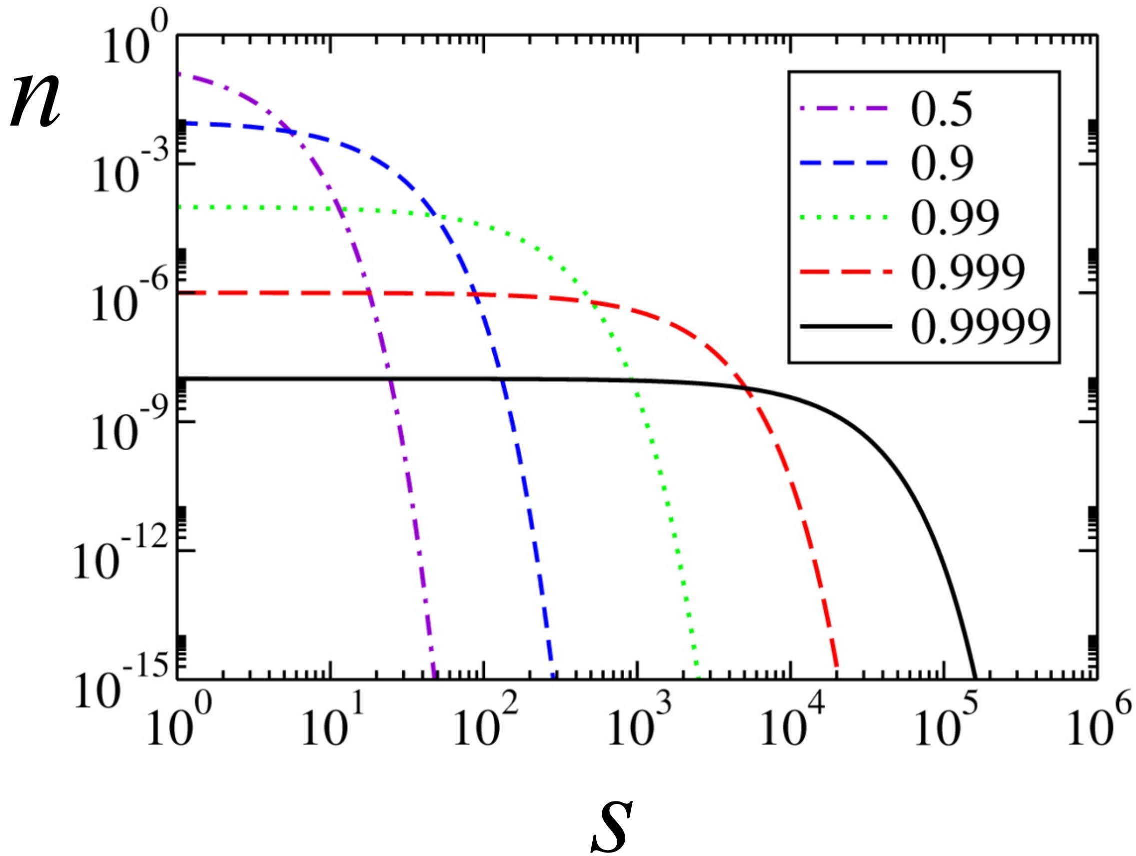
$$n(s,p) = \frac{N(s,p;L)}{L}$$

$$\boxed{n(s,p) = (1-p)^2 p^s}$$

Exact solution in one dimension

cluster number density

$$n(s,p) = (1-p)^2 p^s$$



$$\begin{aligned} n(s,p) &= (1-p)^2 p^s \\ &= (1-p)^2 \exp(\ln p^s) \\ &= (1-p)^2 \exp(s \ln p) \\ &= (1-p)^2 \exp(-s/s_\xi) \end{aligned}$$

$$s_\xi = -\frac{1}{\ln p}$$

$$s_\xi \sim (1-p)^{-1}$$

K. Christensen and N. R. Moloney. *Complexity and Criticality.* Imperial College Press (2005)

Exact solution in one dimension

cluster number density and fluctuations

Probability that a site belongs to a cluster of size s : $sn(s,p) = s(1-p)^2 p^s$

$p < p_c$

$$\sum_s sn(s,p) = \sum_s s(1-p)^2 p^s = p$$

$p > p_c$

$$P_\infty + \sum_{s=1}^{\infty} sn(s,p) = p$$

$p < p_c$

$$\chi(p) = \frac{\sum_s s^2 n(s,p)}{\sum_s sn(s,p)} = \frac{1+p}{1-p}$$

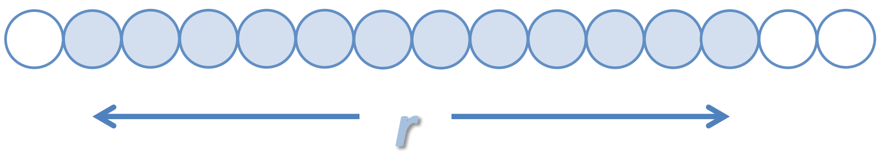
$$\chi(p) \sim (1-p)^{-1}$$

$p < p_c$

Exact solution in one dimension

correlations

Probability that two sites at a distance *r* are connected:

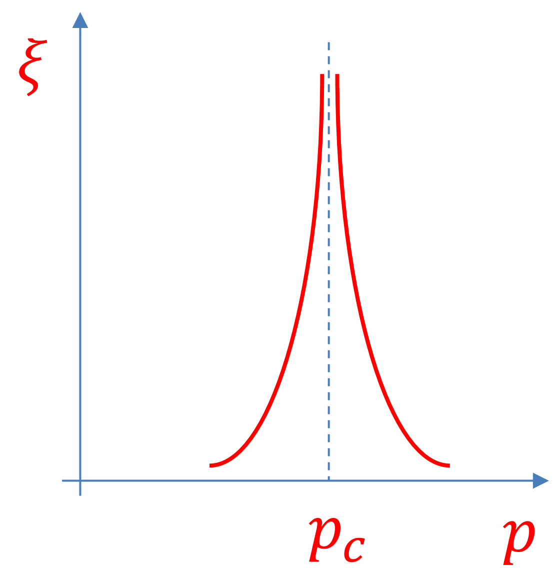


$$g(r) = p^r$$

$$g(r) = \exp\left(-\frac{r}{\xi}\right)$$

$$\xi = -\frac{1}{\ln p}$$

$$\boxed{\xi \sim \frac{1}{(p_c - p)}}$$



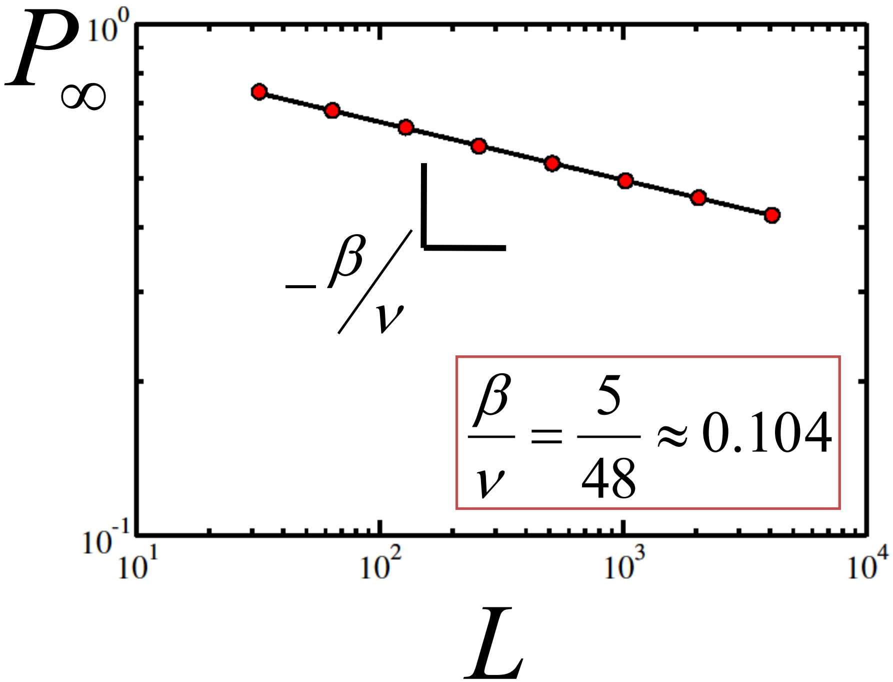
Percolation threshold

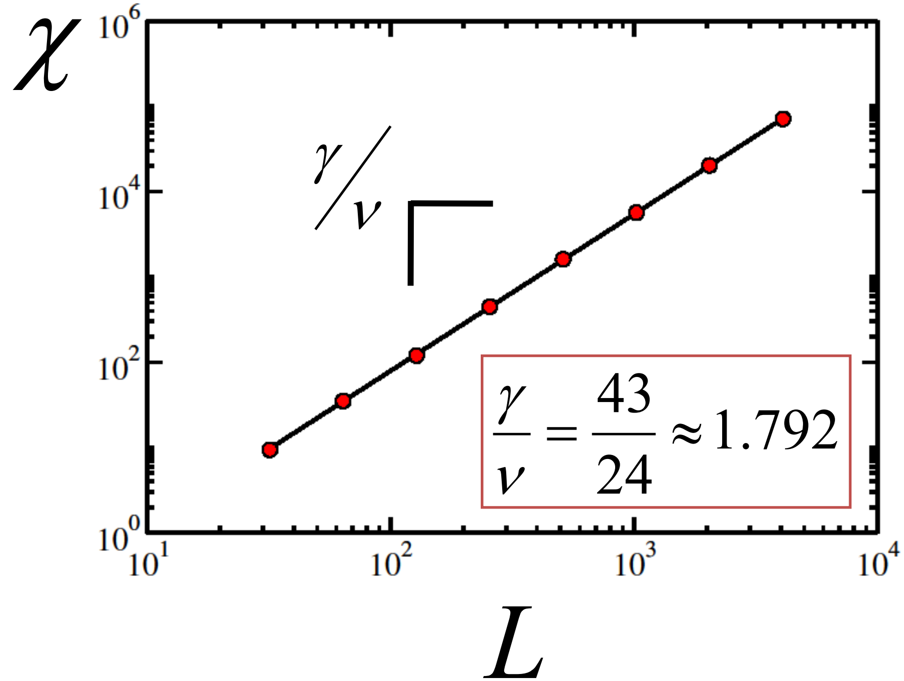
order parameter and fluctuations

$\xi \sim (p_c - p)^{-\nu}$

$$P_\infty \sim (p-p_c)^\beta \sim \begin{cases} \xi^{-\beta/\nu}, & L \gg \xi \\ L^{-\beta/\nu}, & 1 \ll L \ll \xi \end{cases}$$

$$\chi \sim (p-p_c)^{-\gamma} \sim \begin{cases} \xi^{\gamma/\nu}, & L \gg \xi \\ L^{\gamma/\nu}, & 1 \ll L \ll \xi \end{cases}$$





Percolation threshold

finite-size scaling

$$\boxed{\mathcal{O}\left(\ell\varepsilon, \ell^{-\nu}L, \ell^{1/\delta}h\right) = \ell^{a}\mathcal{O}(\varepsilon, L, h)}$$

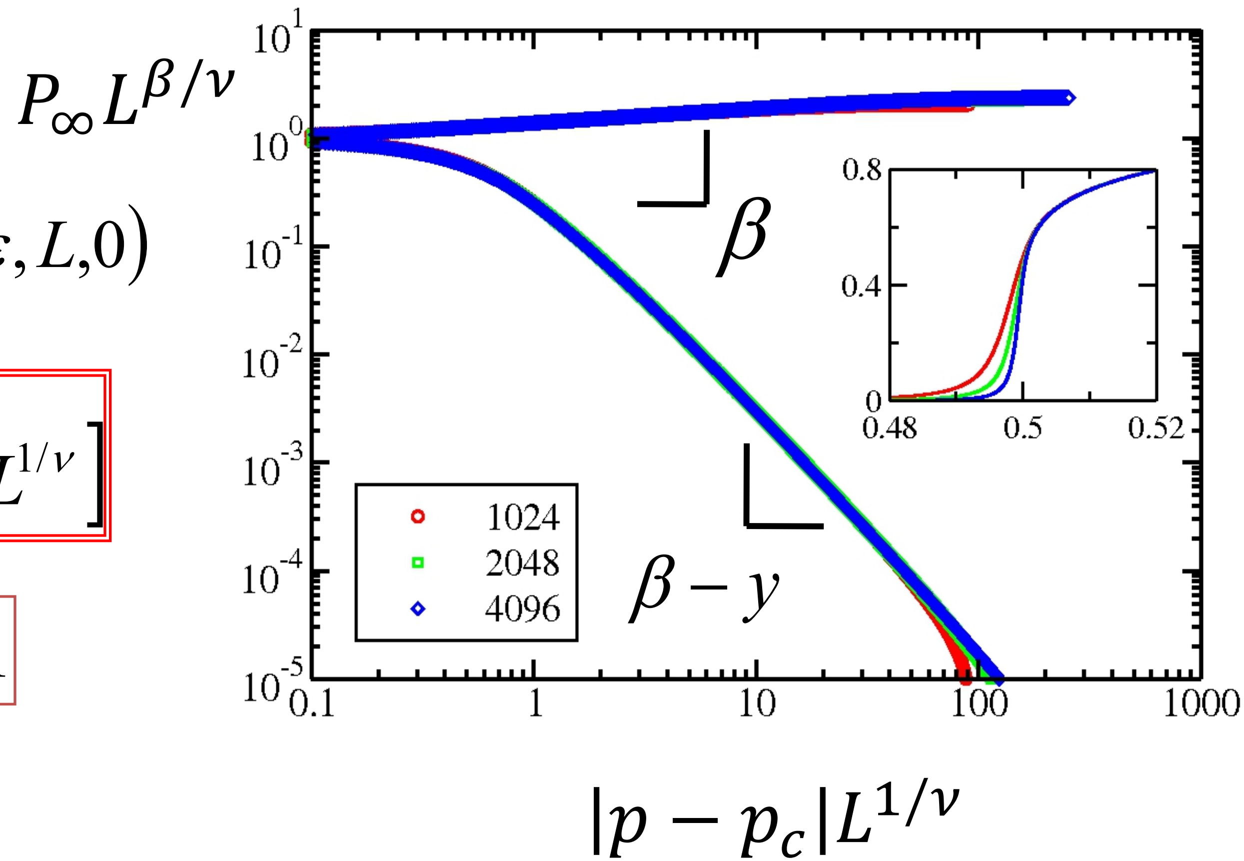
$$\boxed{\varepsilon = p - p_c}$$

$$\ell = L^{1/\nu}, a = \beta, h = 0$$

$$P_\infty\left(\varepsilon L^{1/\nu}, 1, 0\right) = L^{\beta/\nu} P_\infty(\varepsilon, L, 0)$$

$$\boxed{P_\infty = L^{-\frac{\beta}{\nu}} \mathcal{F}\left[(p - p_c)L^{1/\nu}\right]}$$

$$\boxed{\mathcal{F}[x] \sim x^{\beta}, x >> 1}$$



Percolation threshold

finite-size scaling

$$\boxed{\mathcal{O}\left(\ell\varepsilon, \ell^{-\nu}L, \ell^{1/\delta}h\right) = \ell^{a}\mathcal{O}(\varepsilon, L, h)}$$

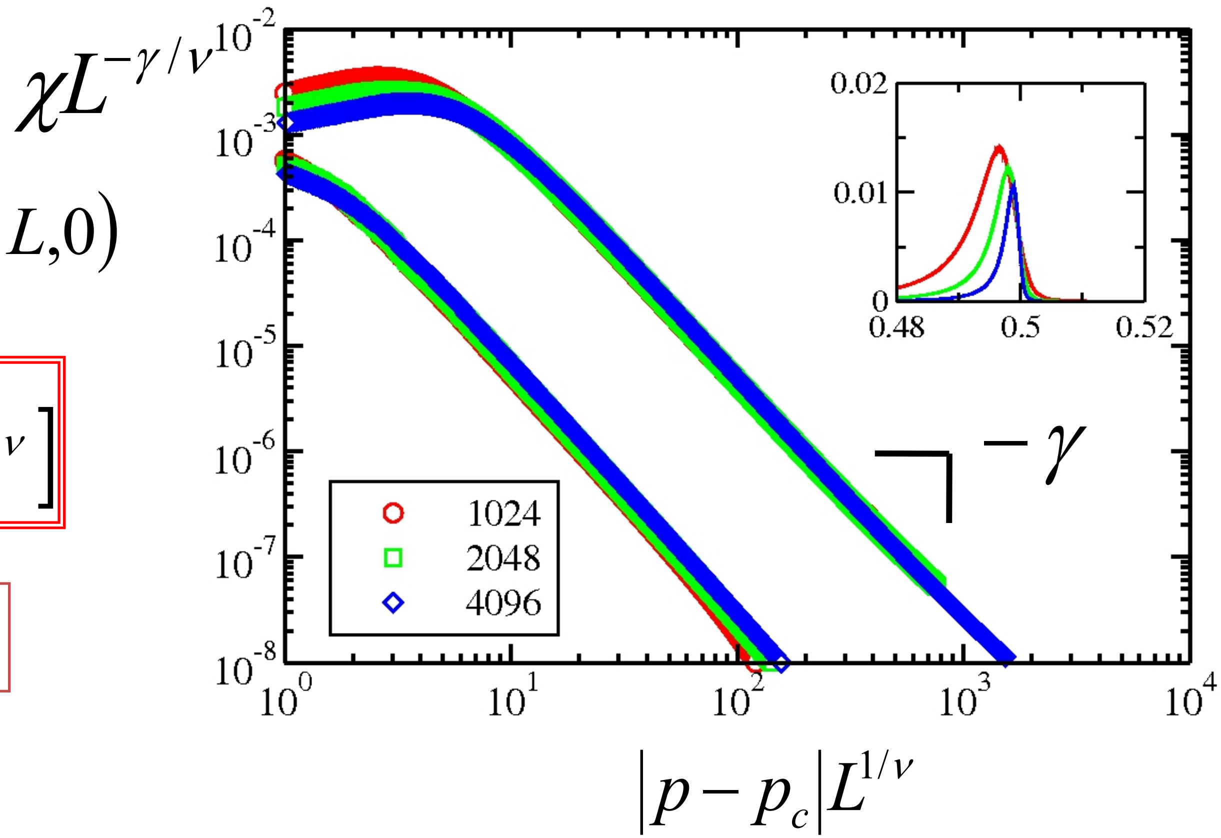
$$\boxed{\varepsilon = p - p_c}$$

$$\ell = L^{1/\nu}, a = -\gamma, h = 0$$

$$\chi\left(\varepsilon L^{1/\nu}, 1, 0\right) = L^{-\gamma/\nu}\chi(\varepsilon, L, 0)$$

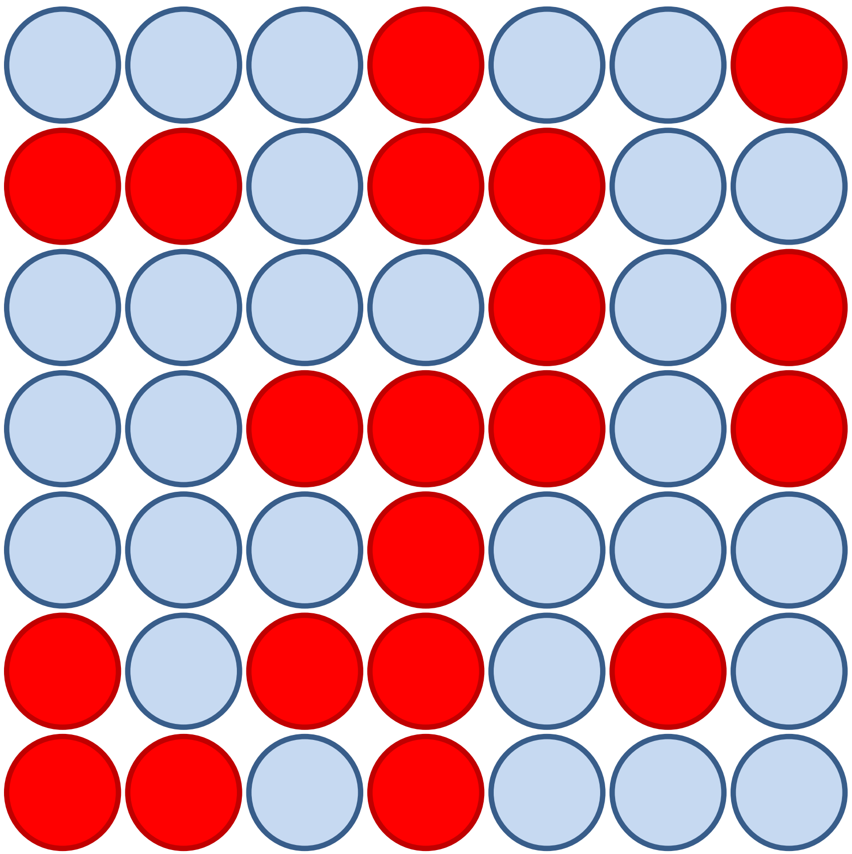
$$\boxed{\chi = L^{\frac{\gamma}{\nu}}\mathcal{F}\left[(p - p_c)L^{1/\nu}\right]}$$

$$\boxed{\mathcal{F}[x] \sim x^{-\gamma}, x >> 1}$$



Algorithms

generate canonical configurations



For each **site *i***:

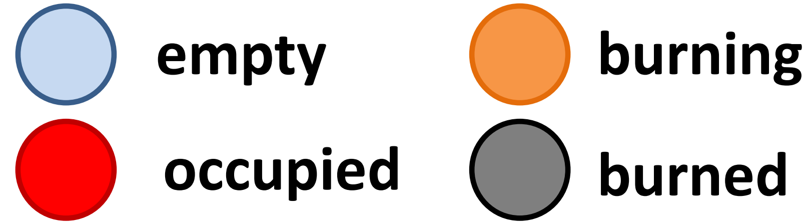
1. **random number ε**;
2. if

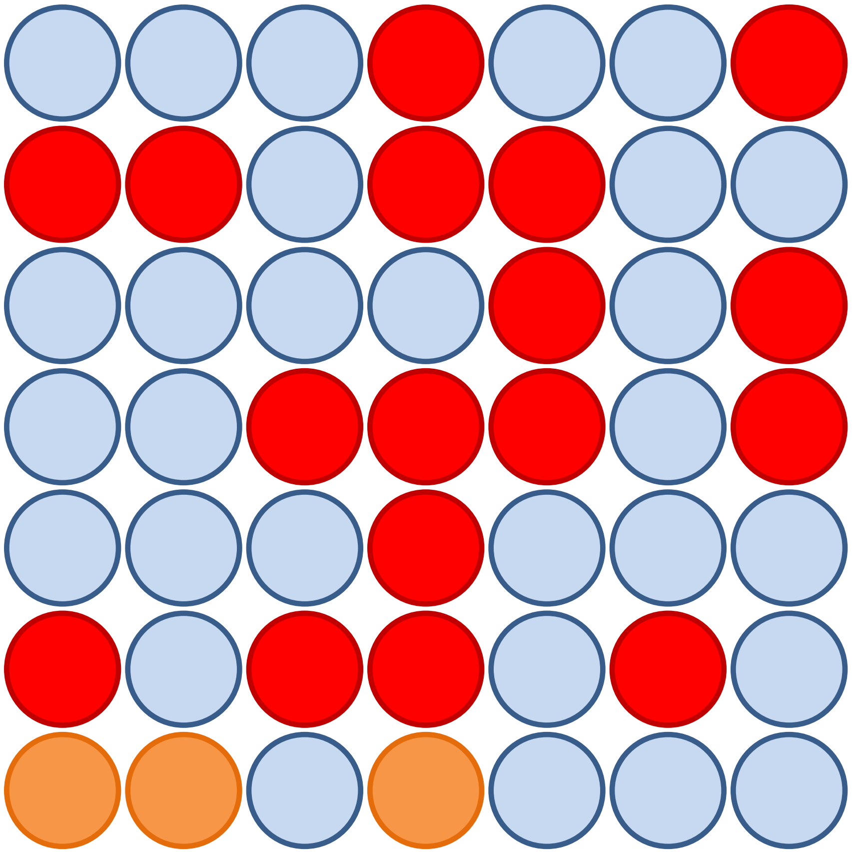
   $\varepsilon < p$: ***i*** is **occupied**;

   else: ***i*** is **empty**.

Algorithms

Burning method

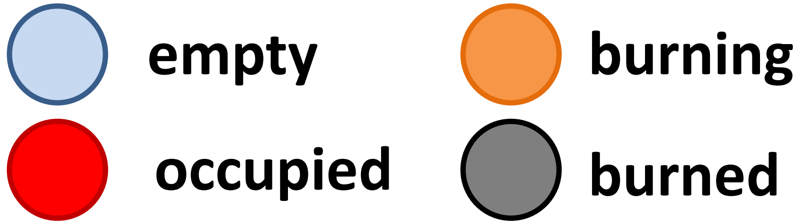


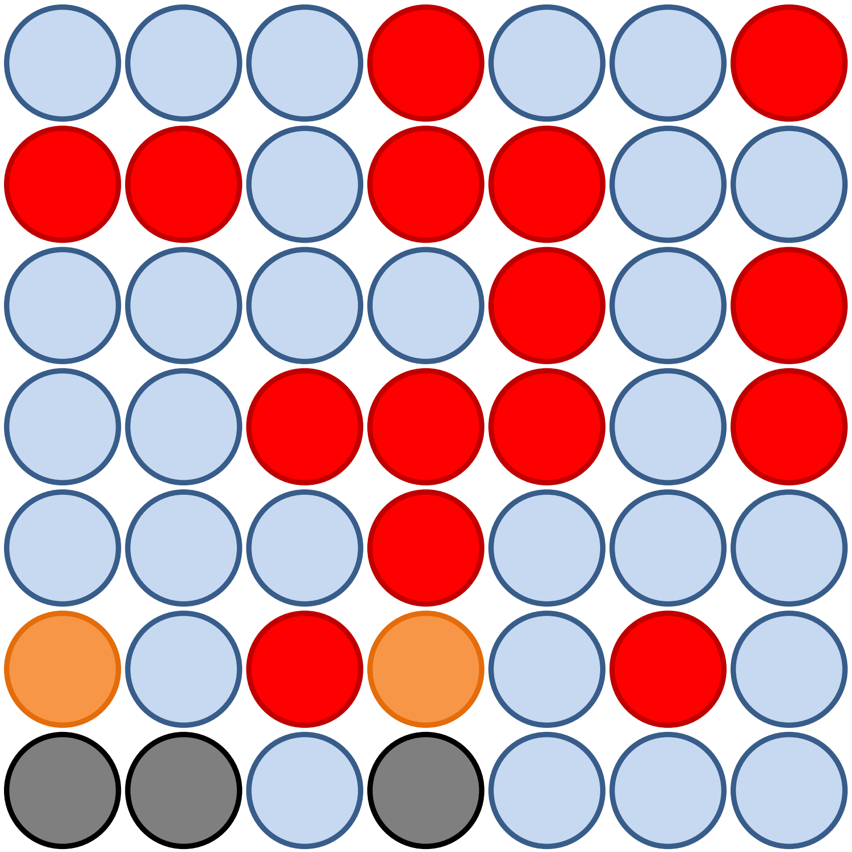


1. **set first row burning**;

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms
Burning method





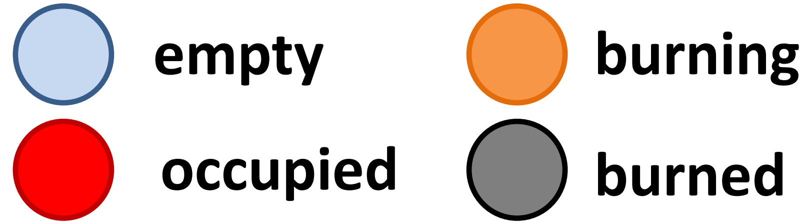
1. **set first** row **burning**;

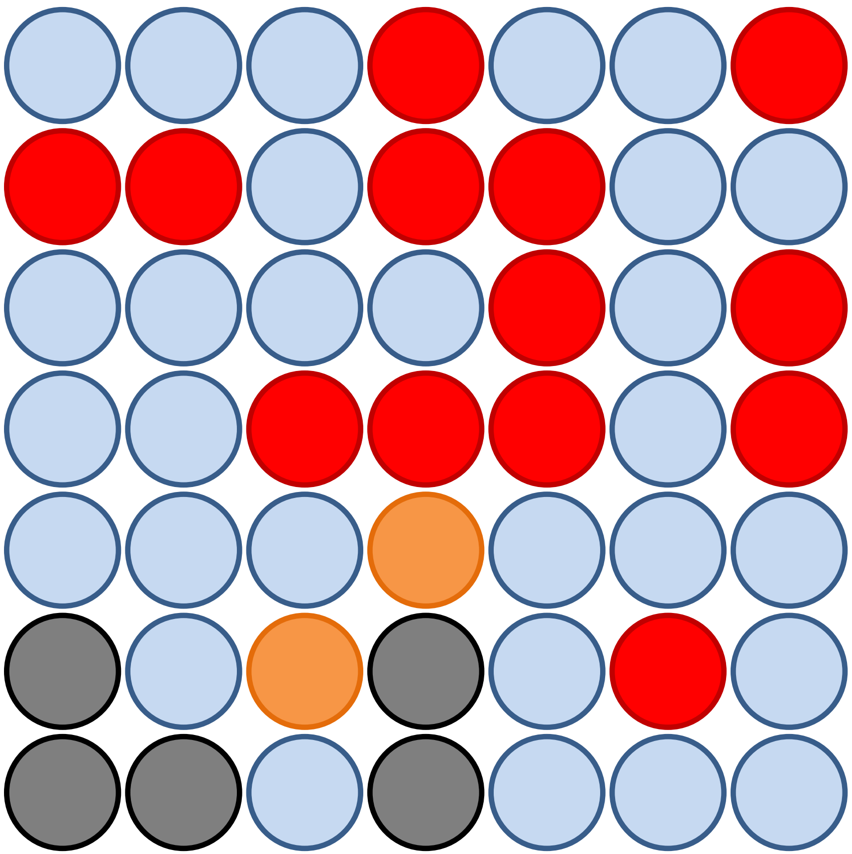
2. **set neighbors of burning** to **burning** and **burning** to **burned**;

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



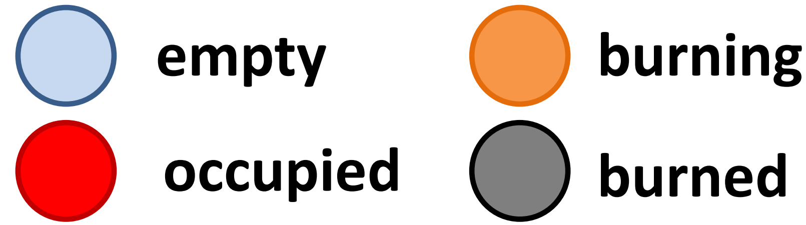


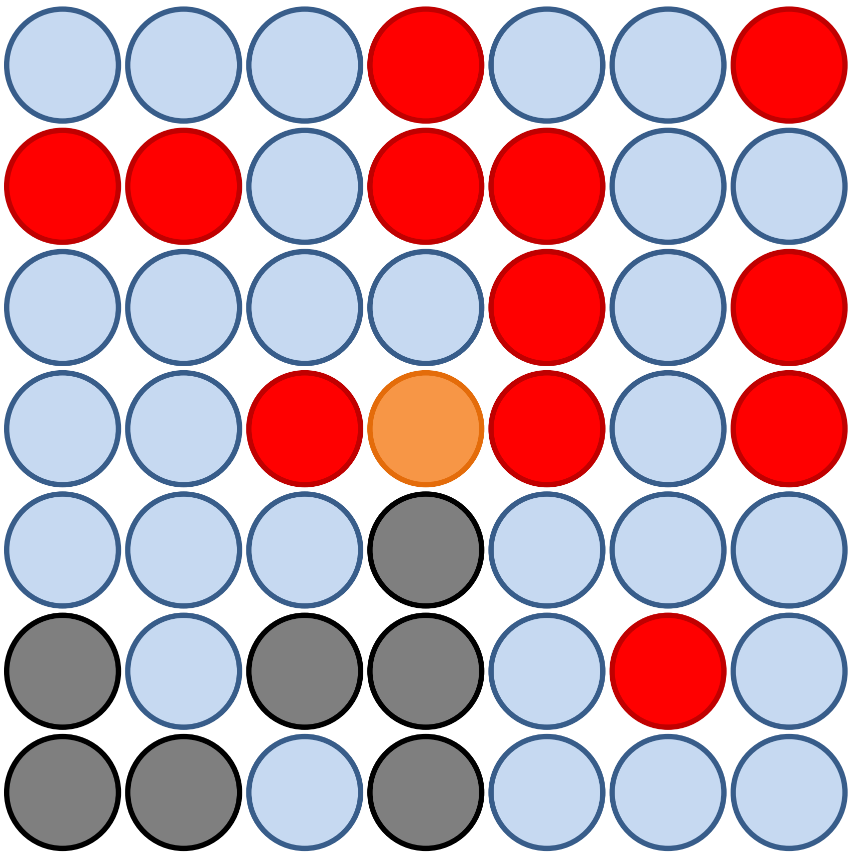
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



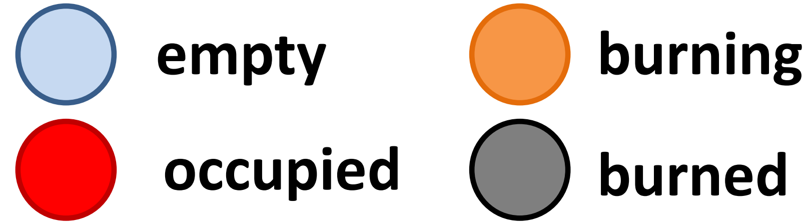


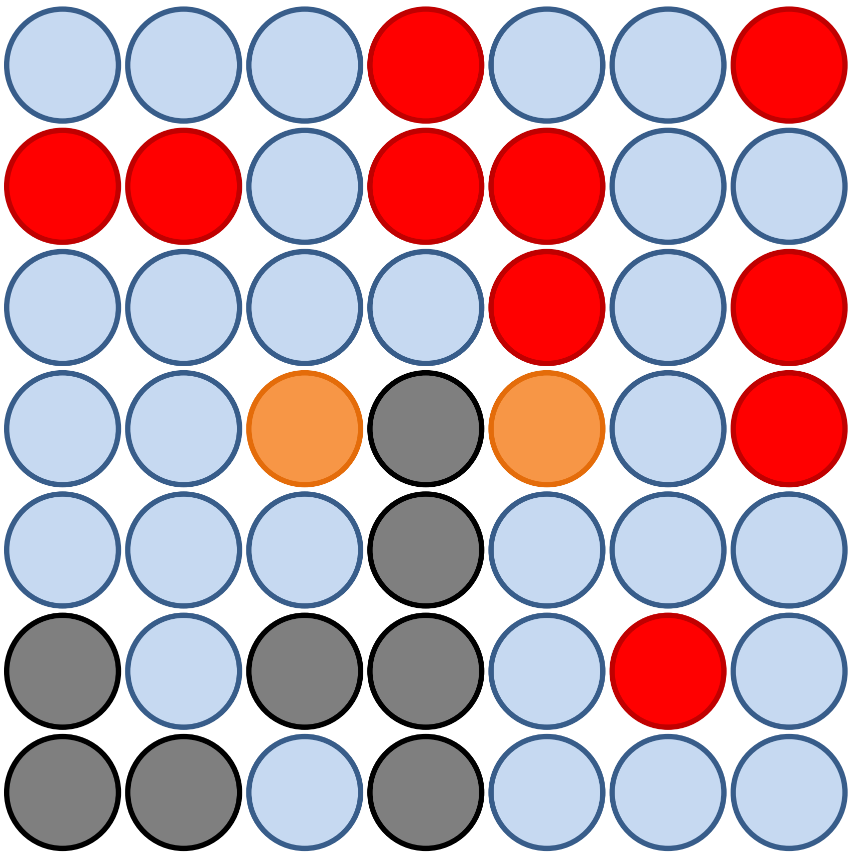
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

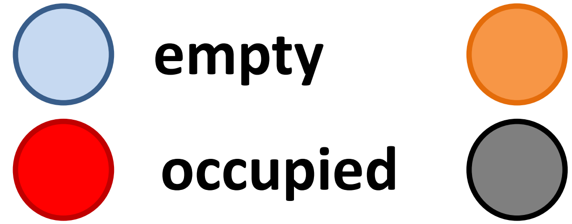


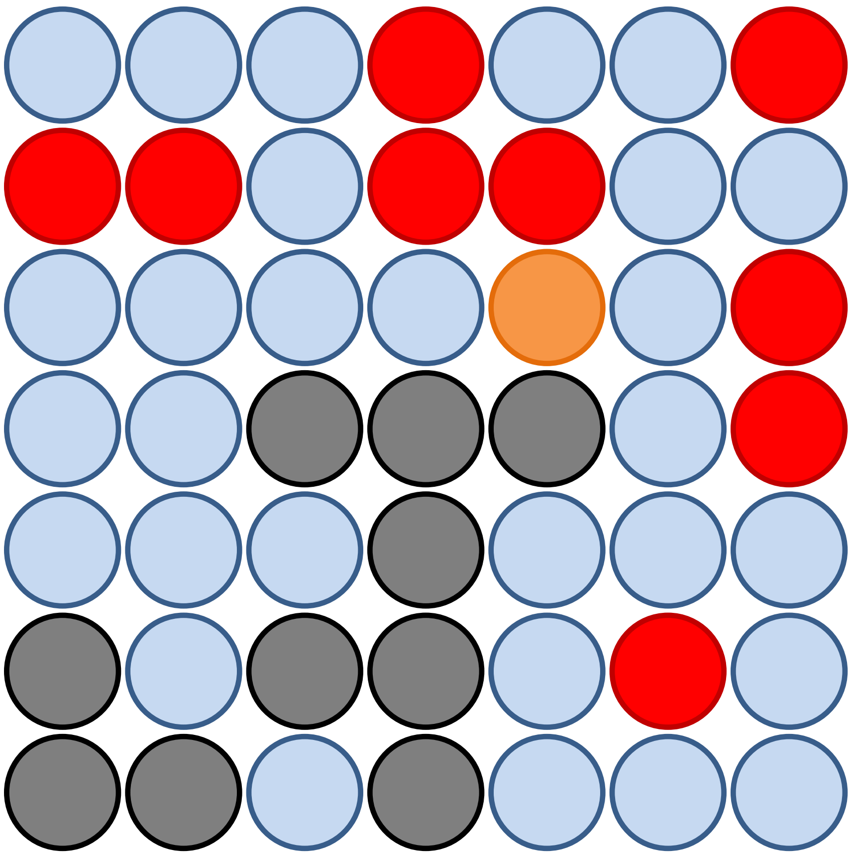


1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms
Burning method



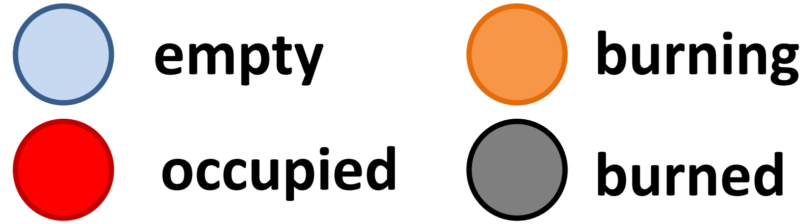


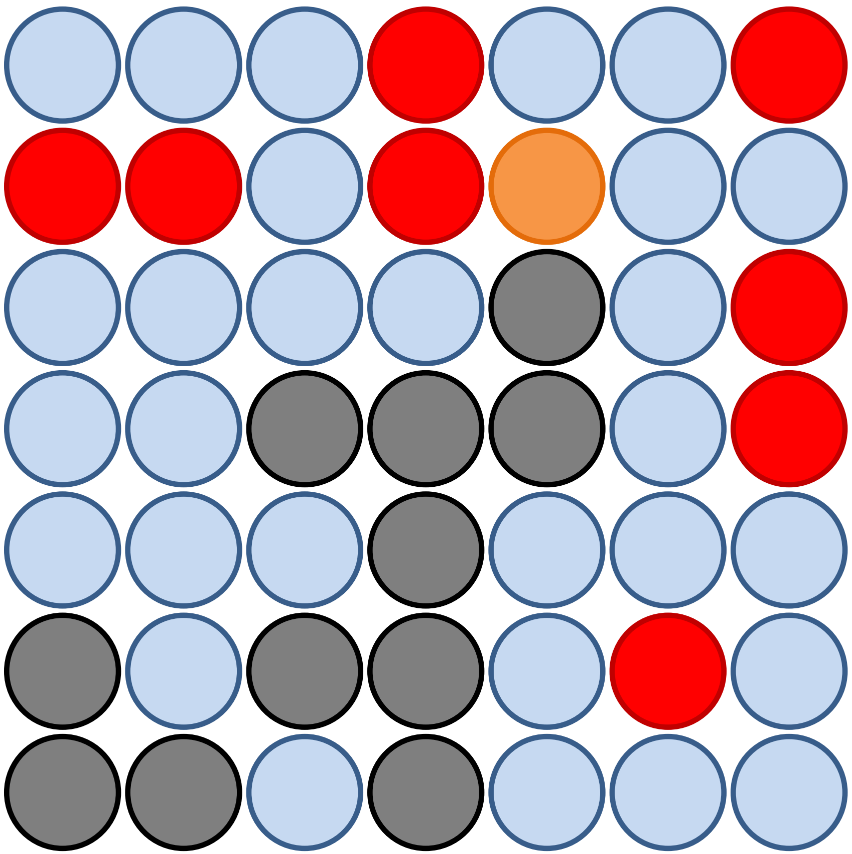
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



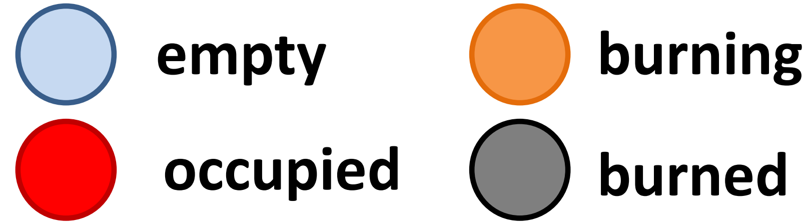


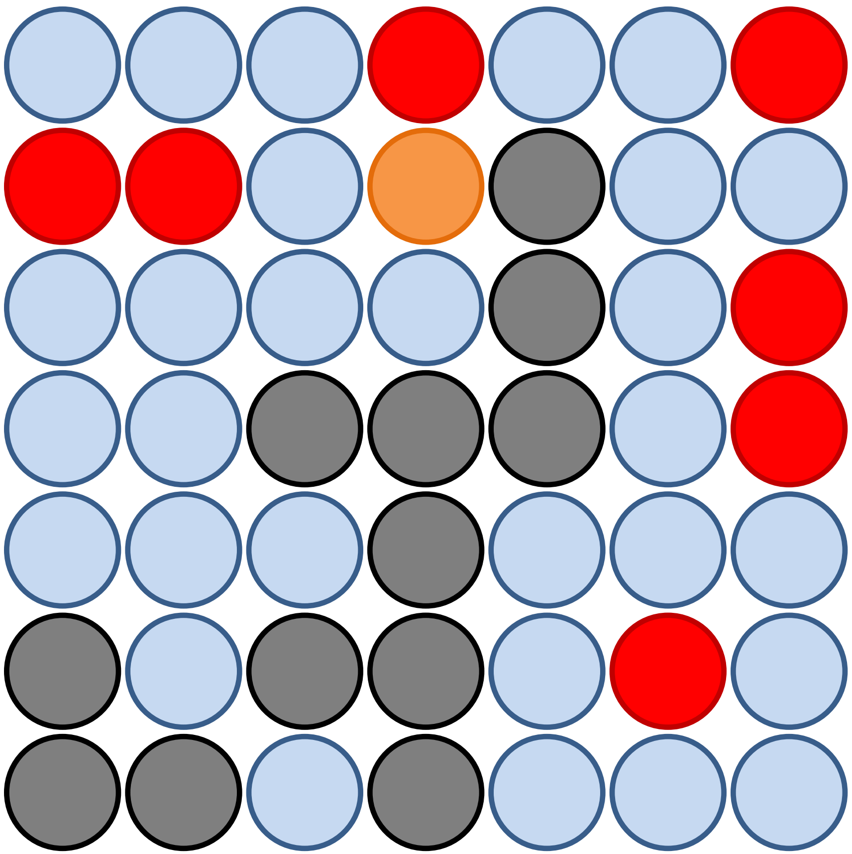
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



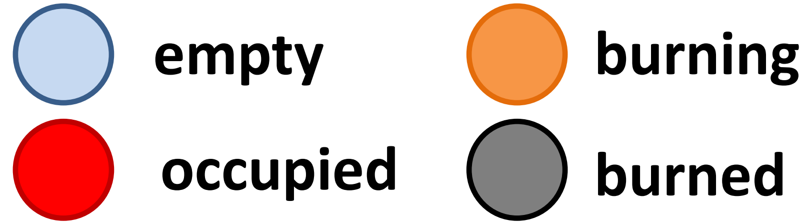


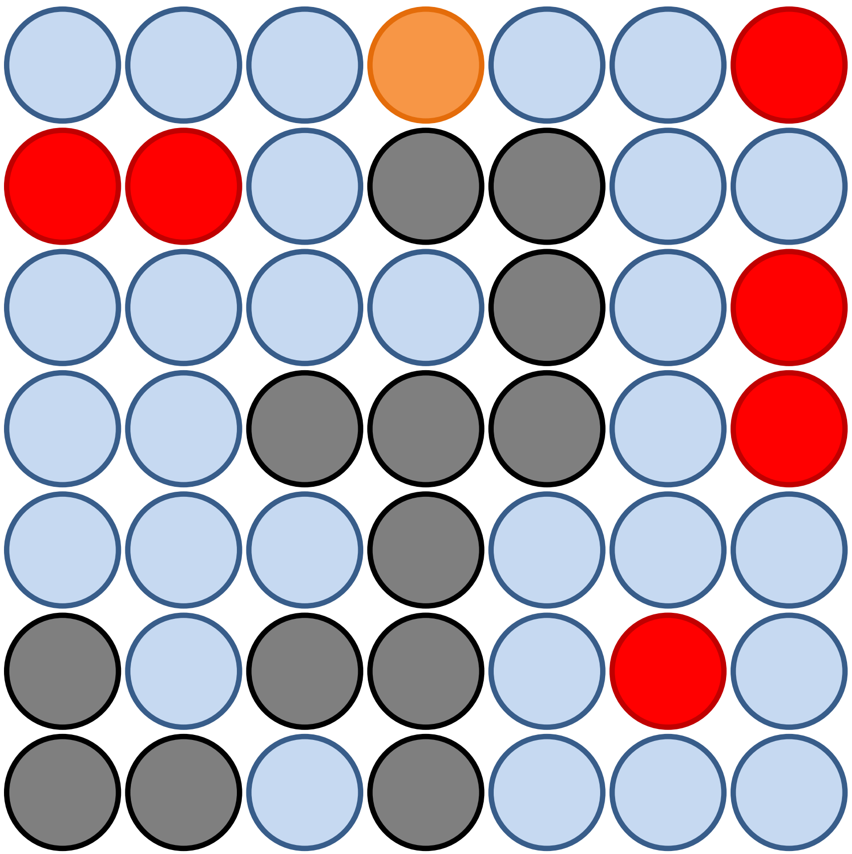
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



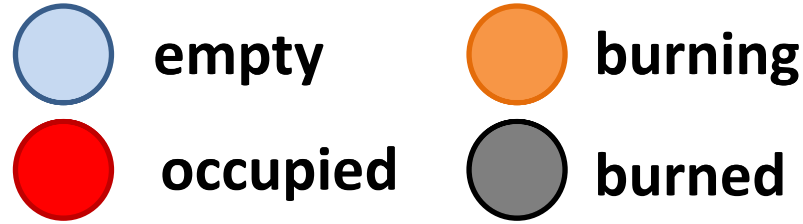


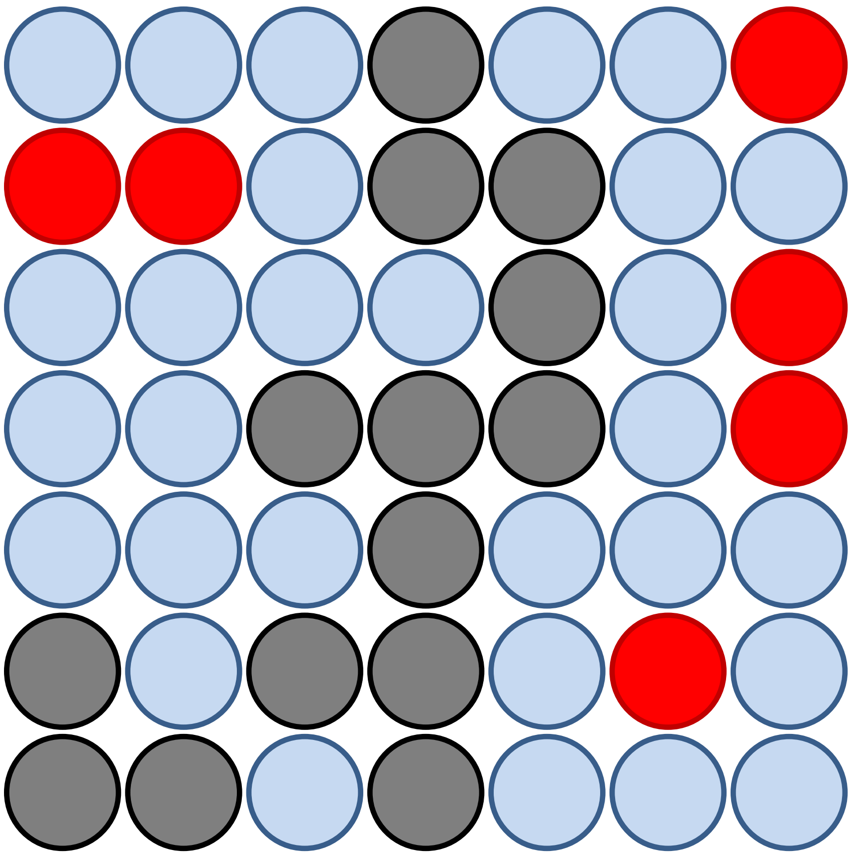
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method



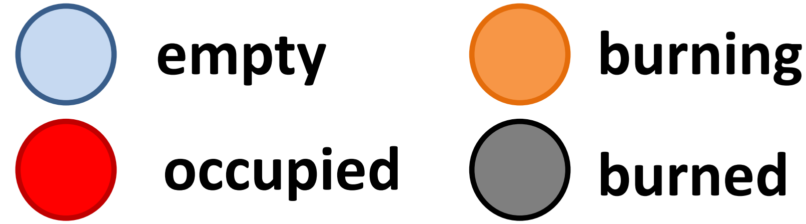


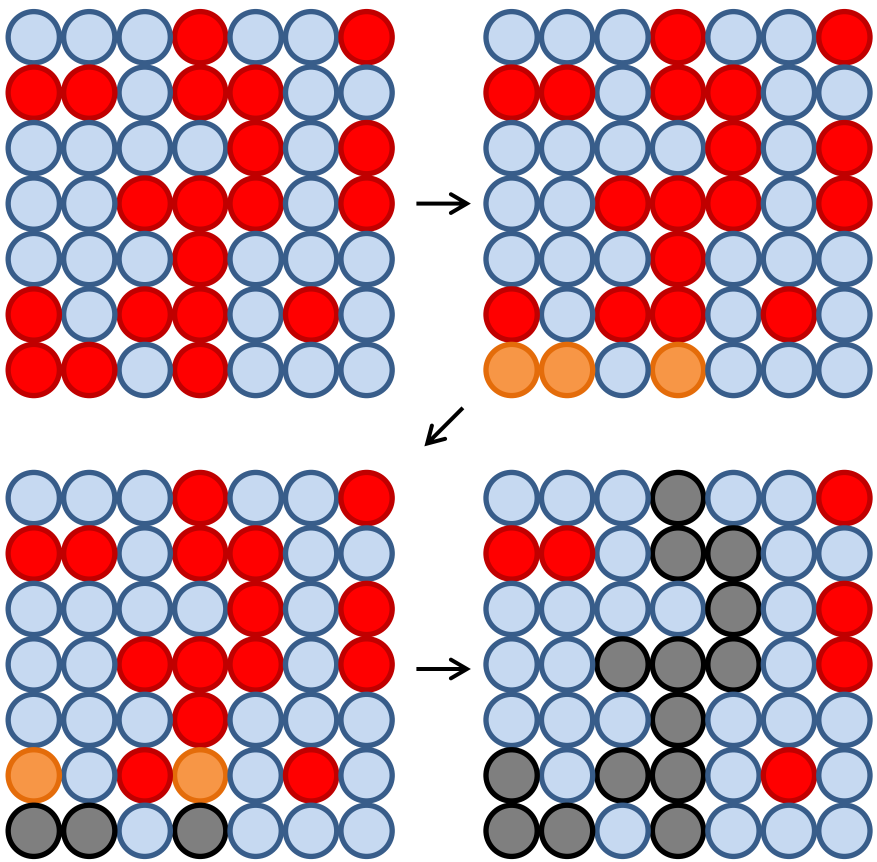
1. **set first** row **burning**;
2. **set neighbors of burning** to **burning** and **burning** to **burned**;
3. **repeat** until everything is **burned**.

H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method





One can determine if the set of occupied sites percolates or not.

Number of clusters and cluster-size distribution?

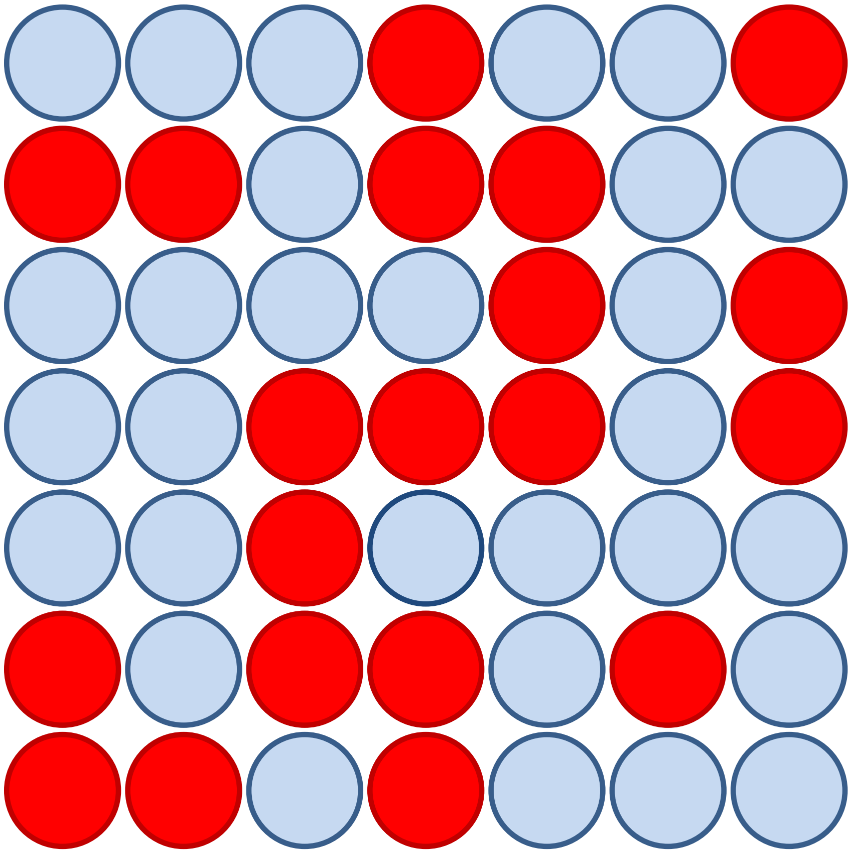
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Hoshen and Kopelman

k = 2

M(k) = 0



1. **start** from the site in the **left-bottom corner**;
2. **sweep** from **left to right bottom to top**;
3. **only** verify **left** and **bottom** neighbors.

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

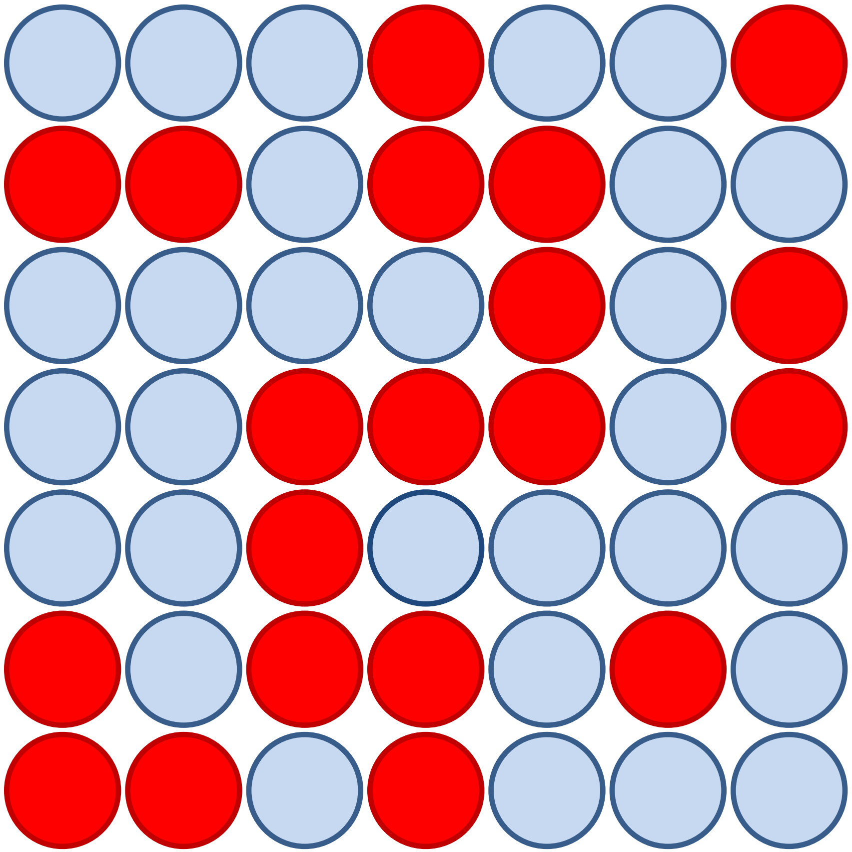
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	0

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

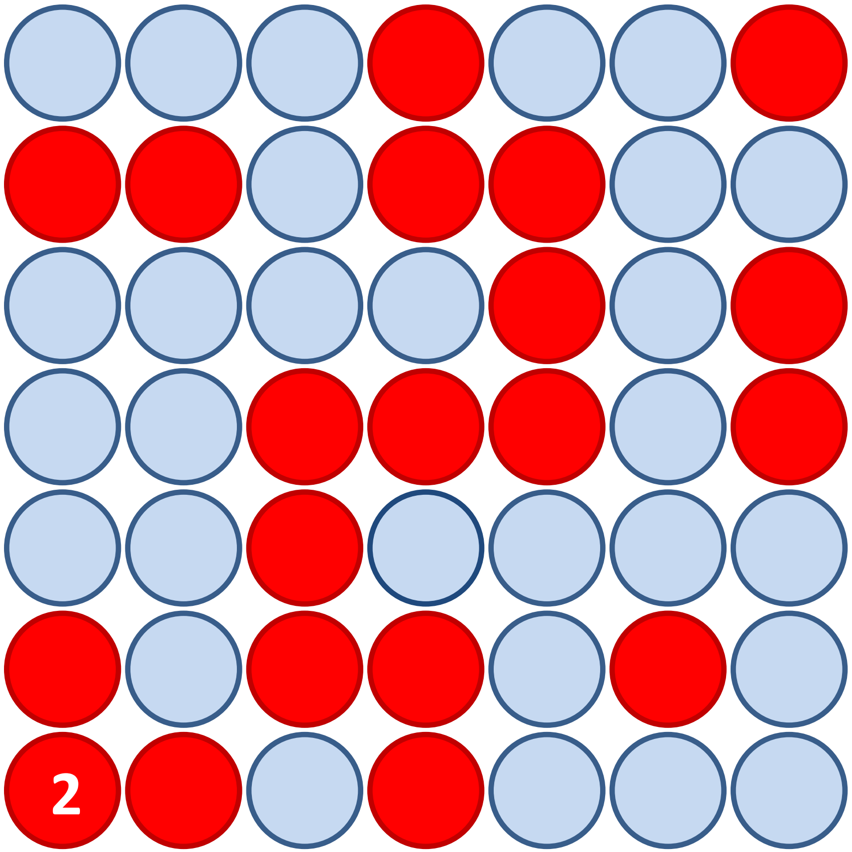
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	1

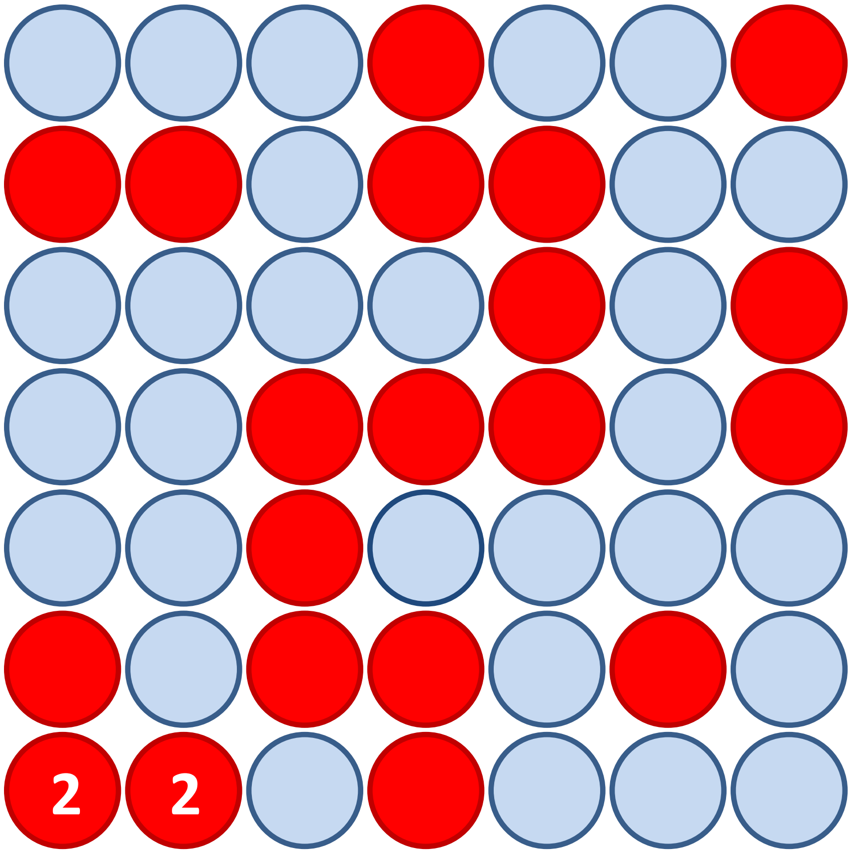
J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

Algorithms
Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	2

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

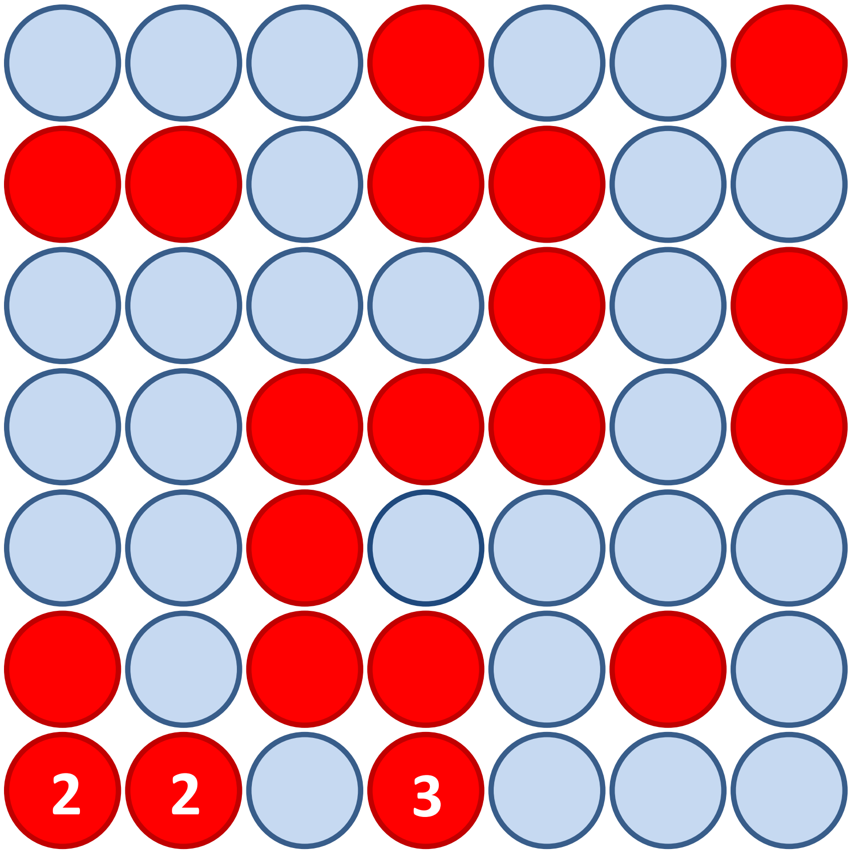
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	2
3	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

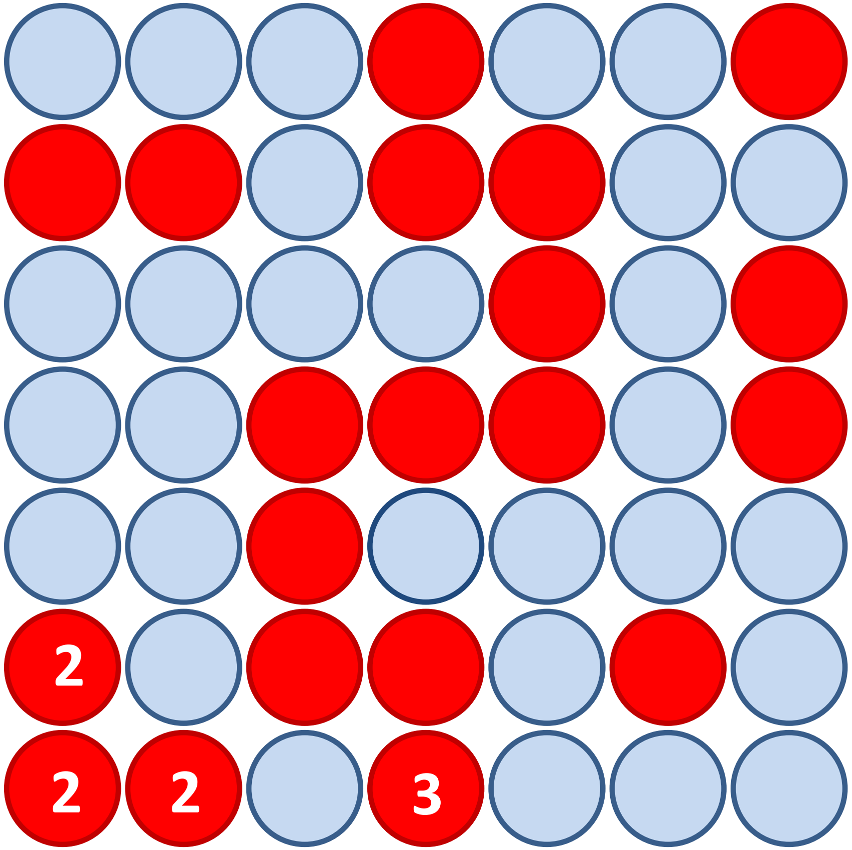
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

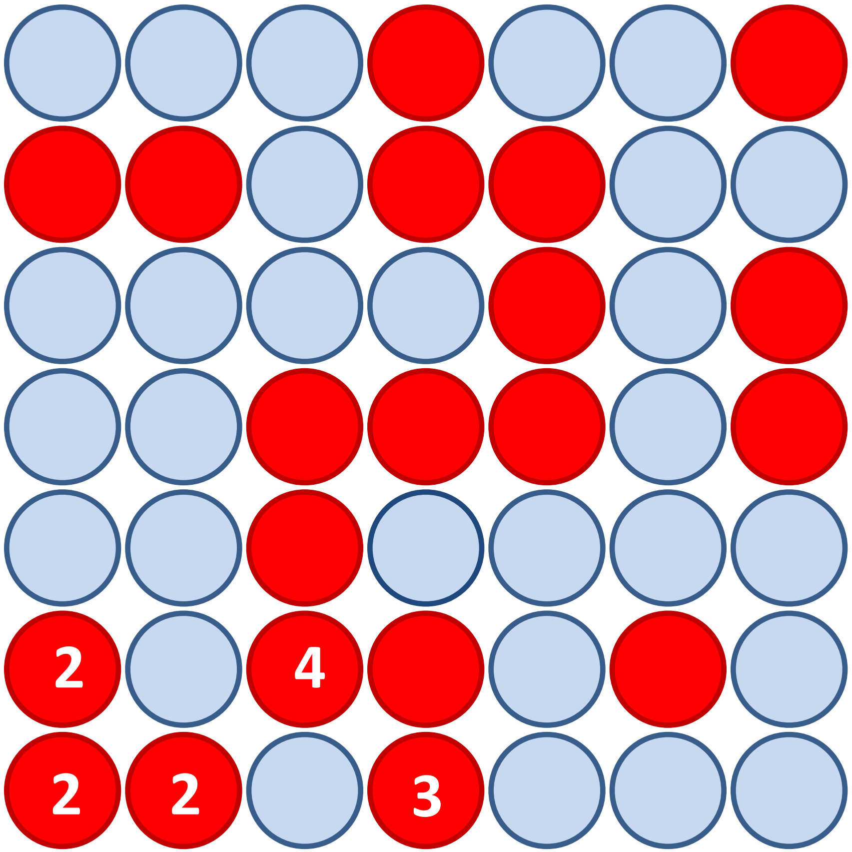
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	1
4	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

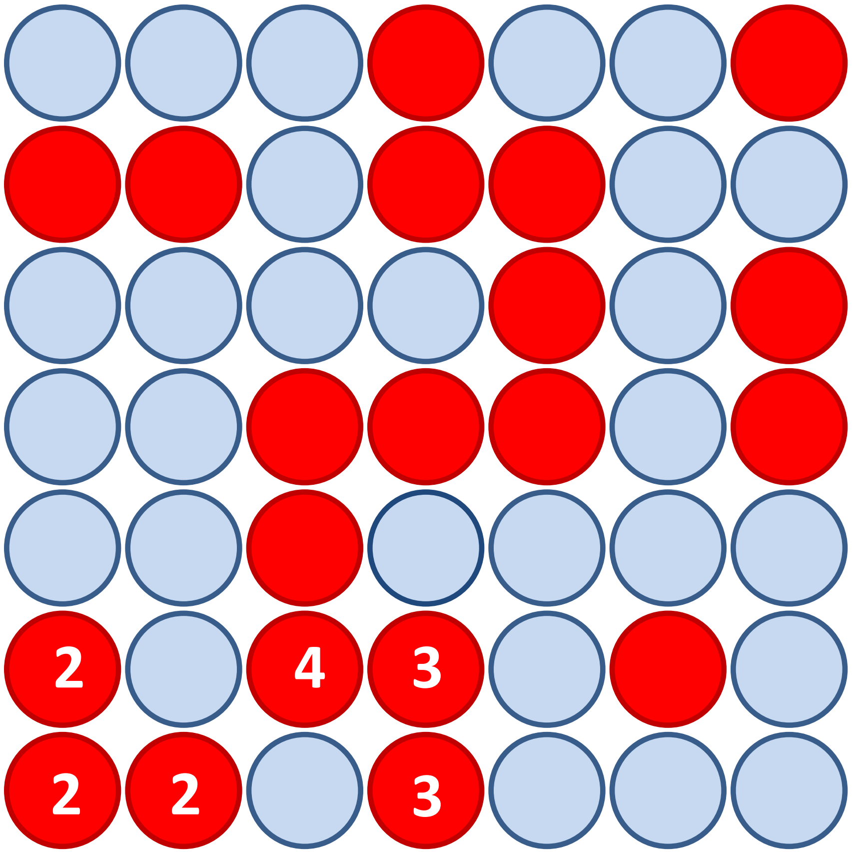
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	3
4	-3

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

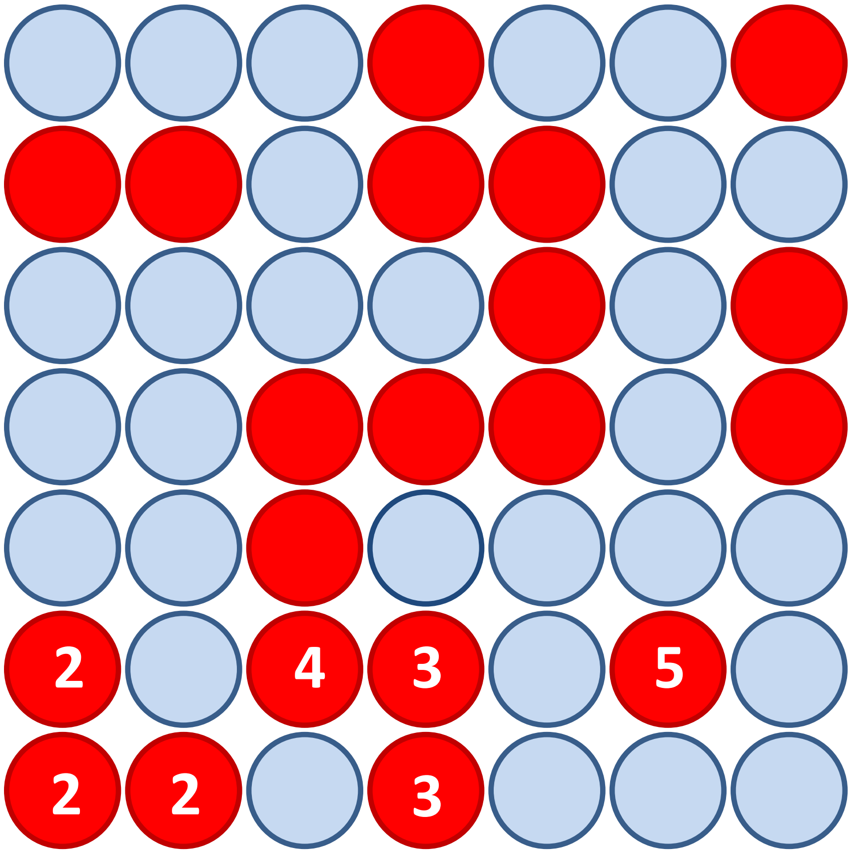
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	3
4	-3
5	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

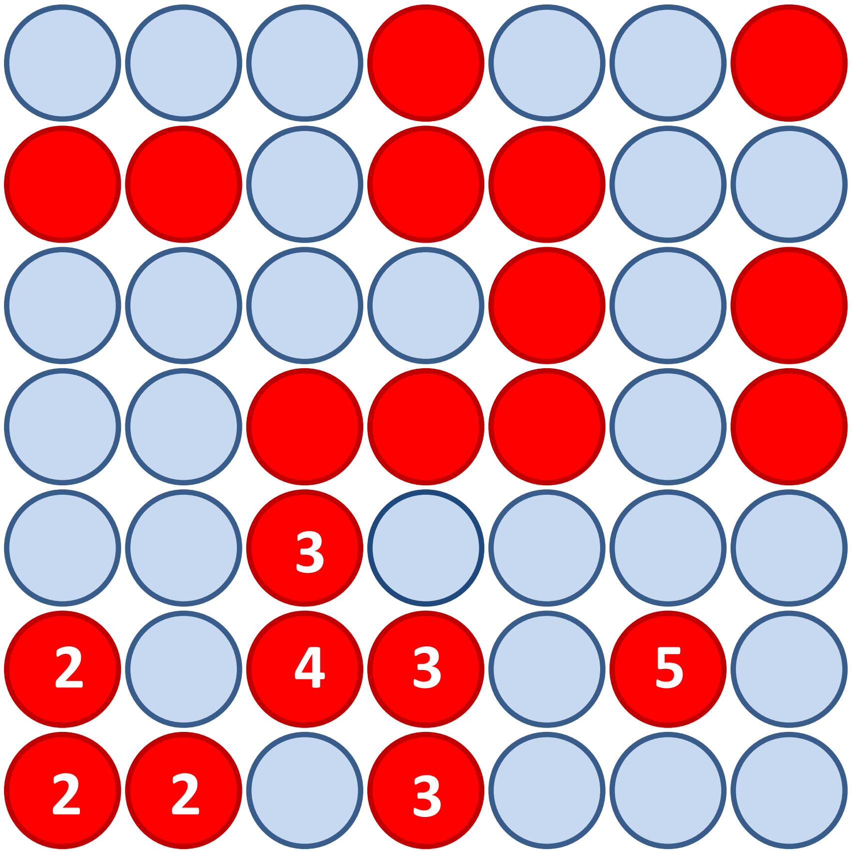
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	4
4	-3
5	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

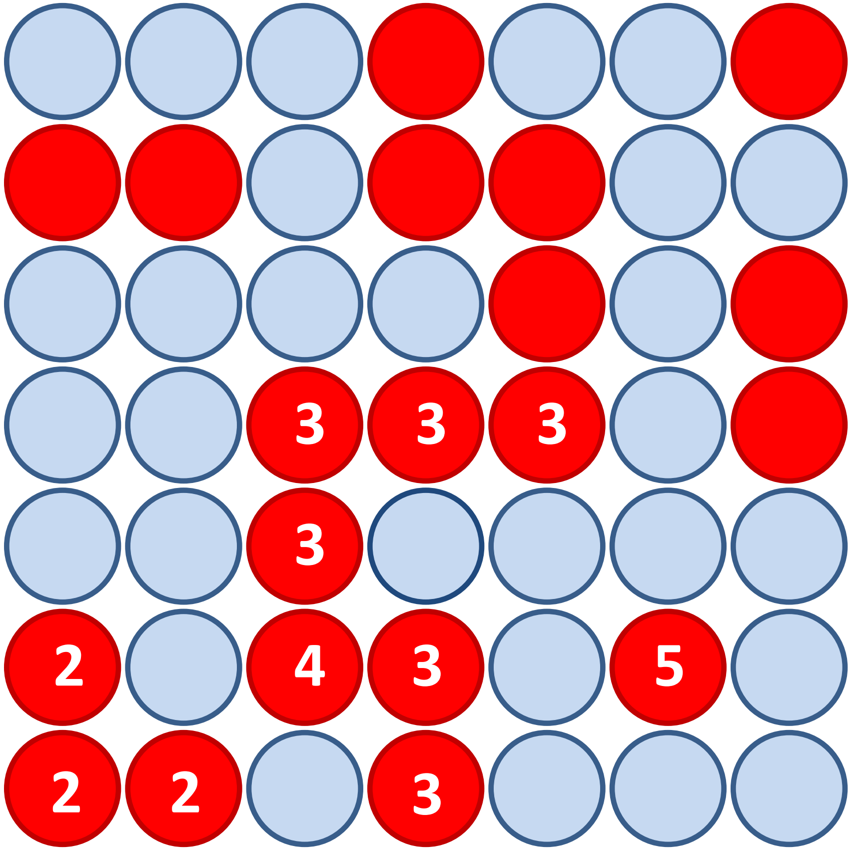
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	7
4	-3
5	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

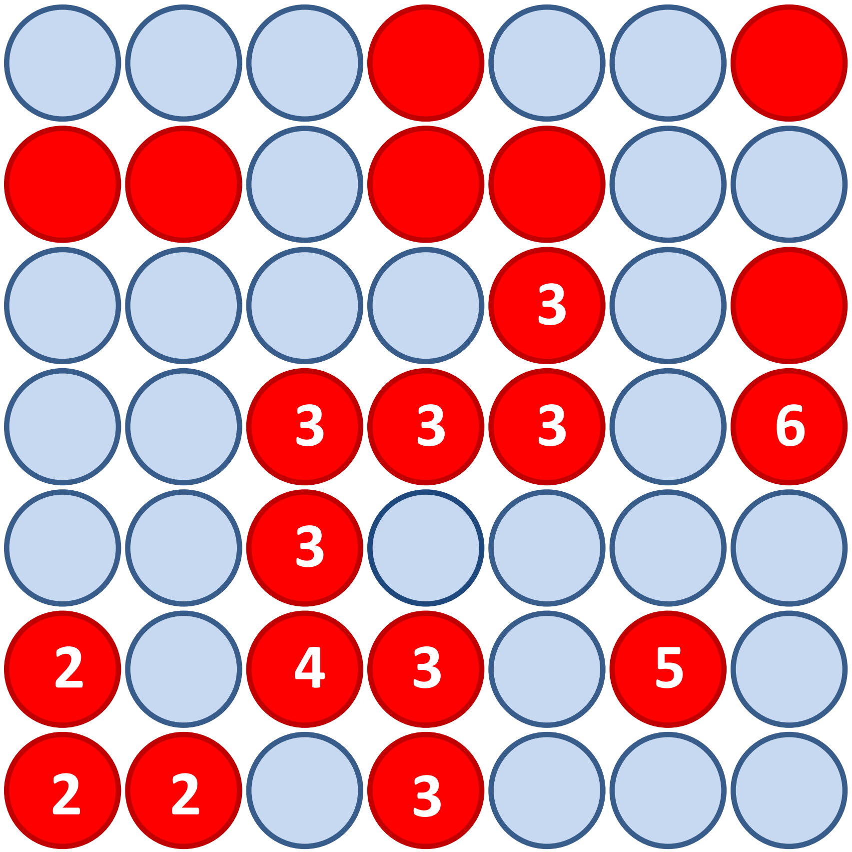
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	8
4	-3
5	1
6	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

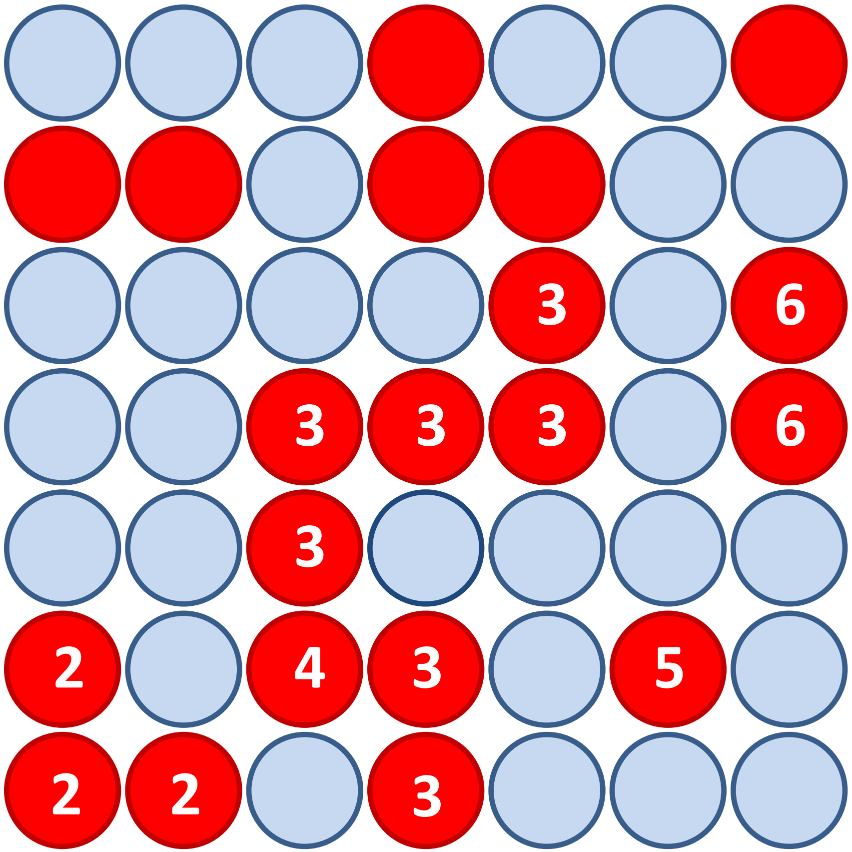
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	8
4	-3
5	1
6	2

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

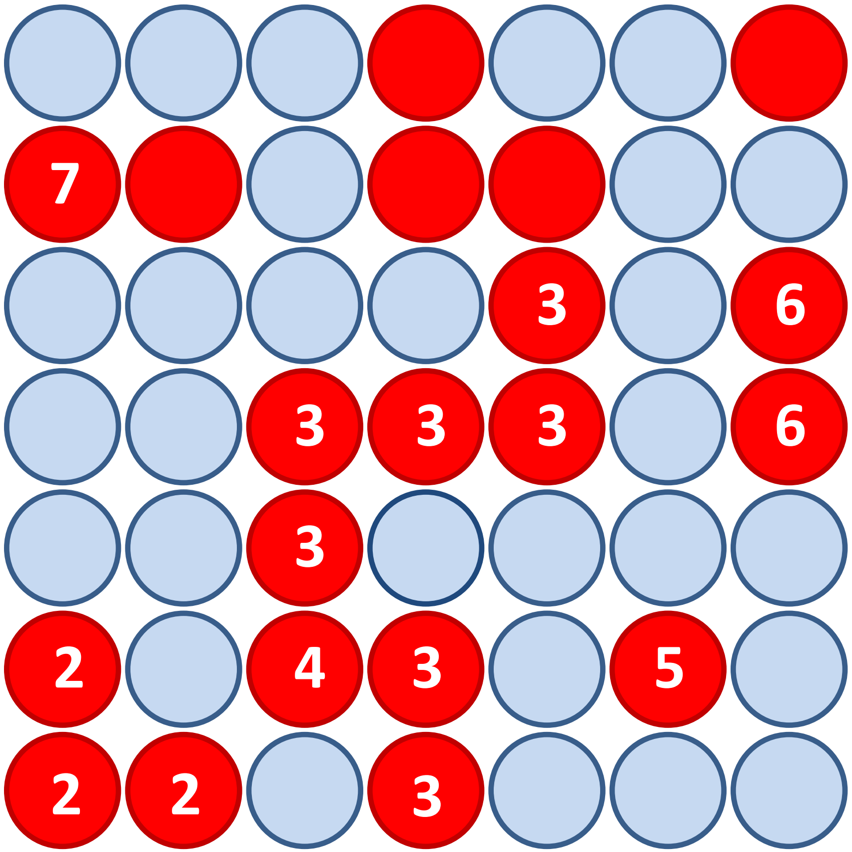
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	8
4	-3
5	1
6	2
7	1

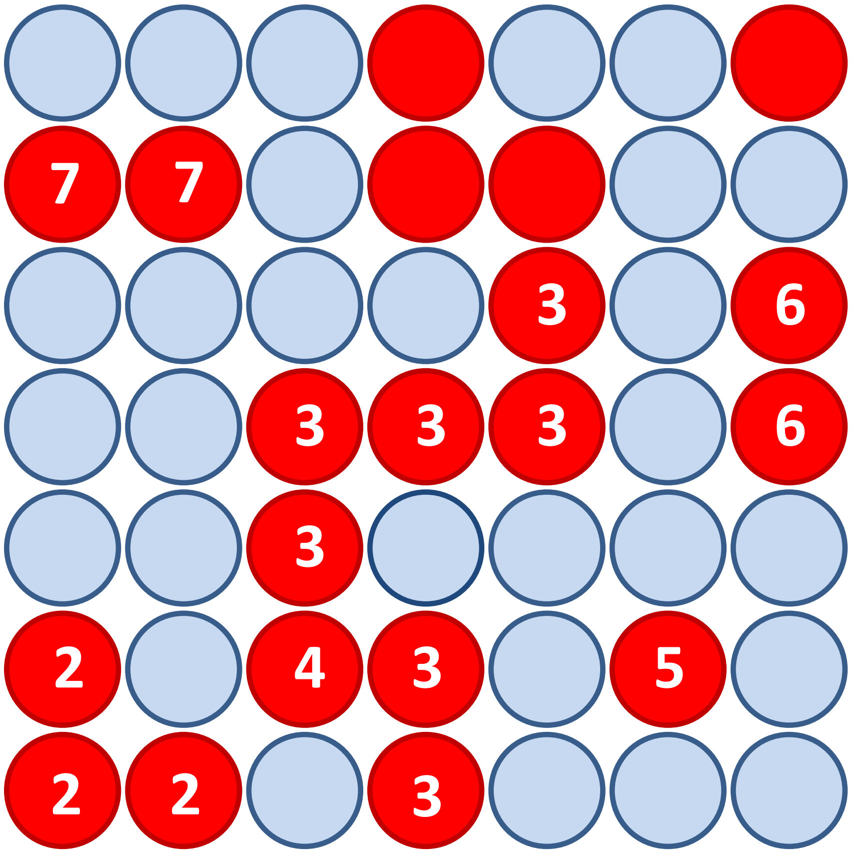
J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

Algorithms
Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	8
4	-3
5	1
6	2
7	2

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

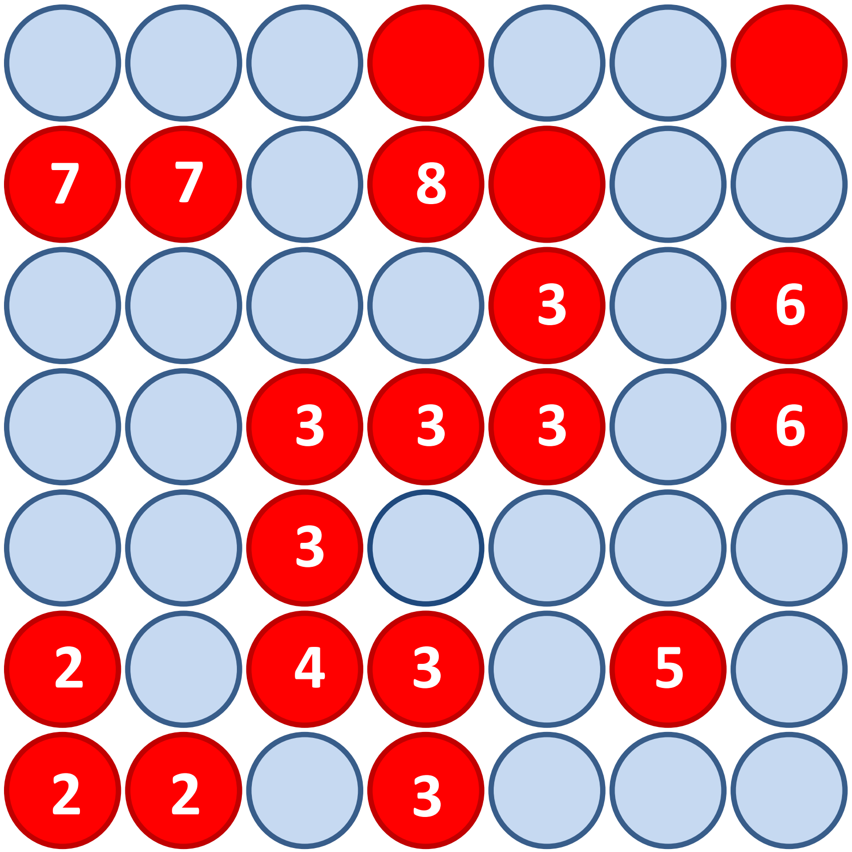
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	8
4	-3
5	1
6	2
7	2
8	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

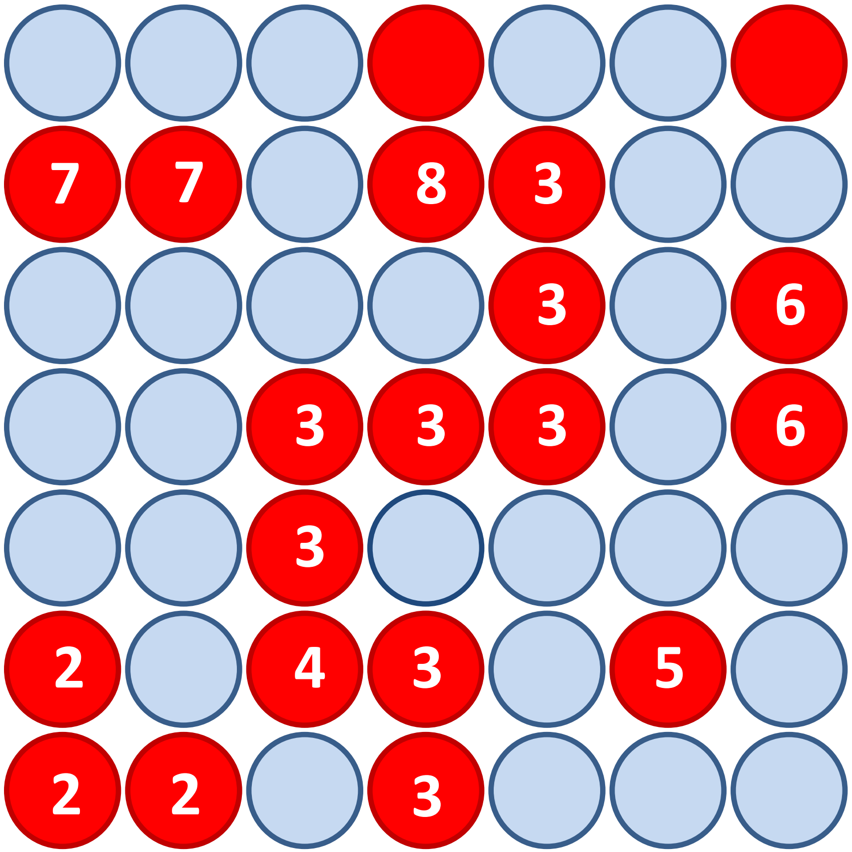
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	10
4	-3
5	1
6	2
7	2
8	-3

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

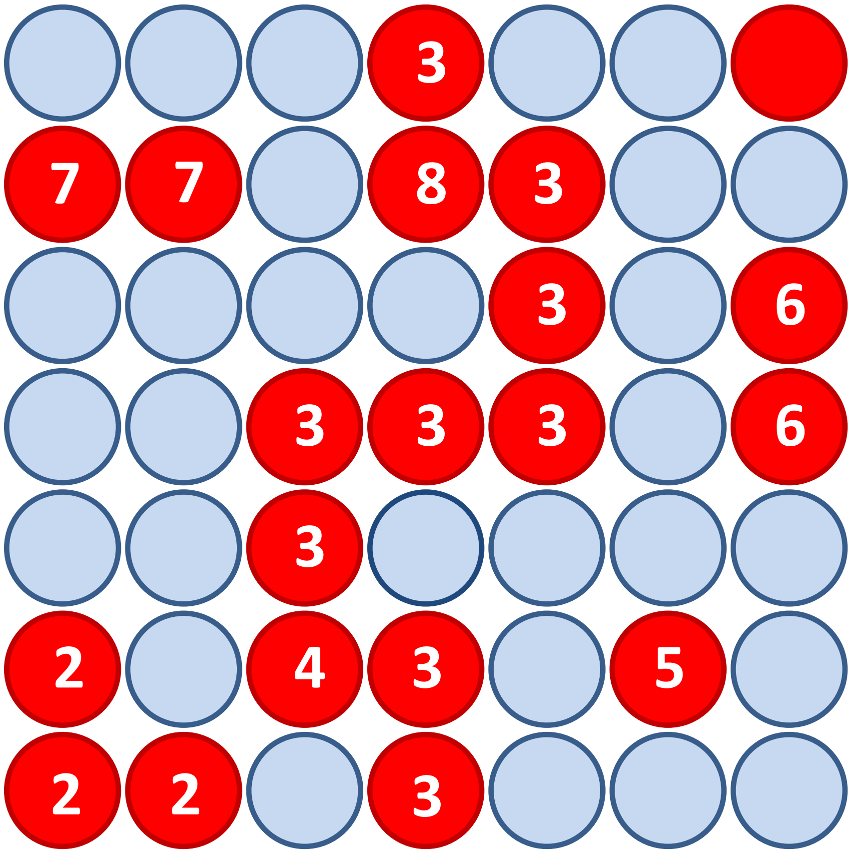
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.



k	M(k)
2	3
3	11
4	-3
5	1
6	2
7	2
8	-3

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

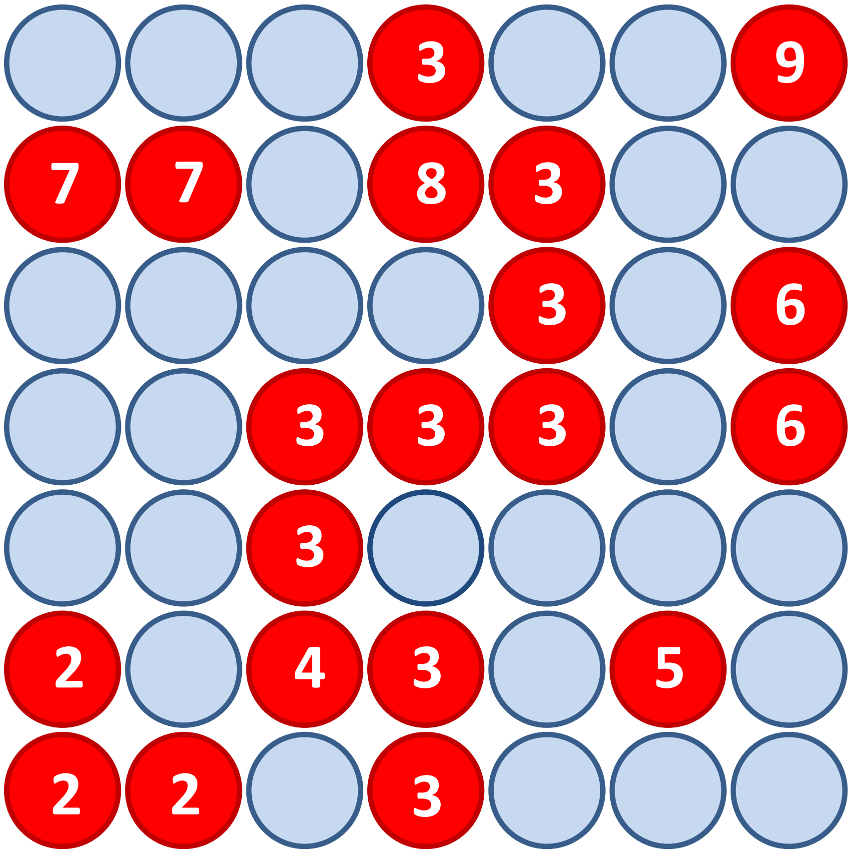
Algorithms

Hoshen and Kopelman

1. **start** from the site in the **left-bottom corner**;

2. **sweep** from **left to right bottom to top**;

3. **only** verify **left** and **bottom** neighbors.

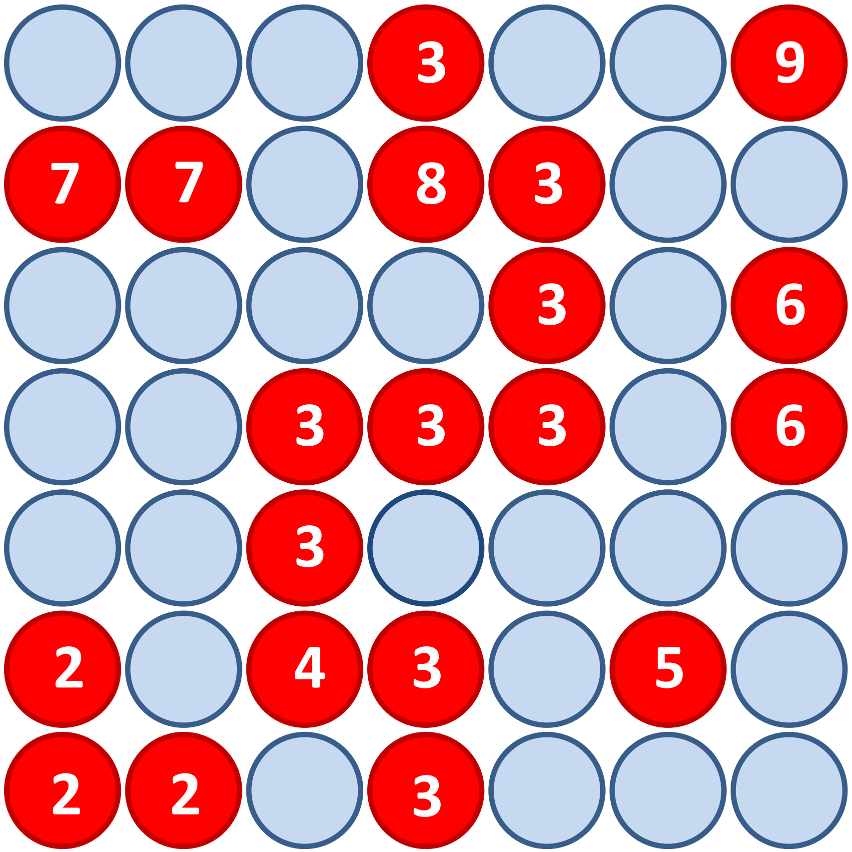


k	M(k)
2	3
3	11
4	-3
5	1
6	2
7	2
8	-3
9	1

J. Hoshen an… 8 (1976)

Algorithms

Hoshen and Kopelman

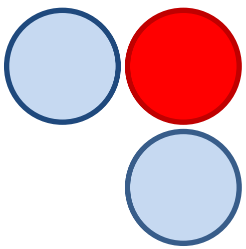


k	M(k)
2	3
3	11
4	-3
5	1
6	2
7	2
8	-3
9	1

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

Algorithms

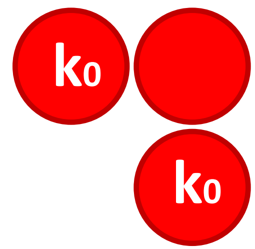
Hoshen and Kopelman



Isolated

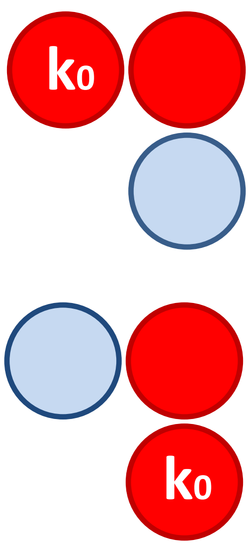
$k = k + 1$

$M(k) = 1$



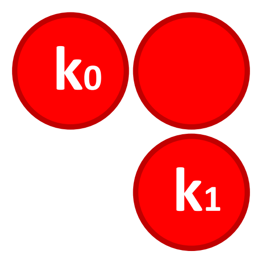
Two neighbor k_0:

$M(\underline{k_0}) = M(\underline{k_0}) + 1$



One neighbor k_0:

$M(\underline{k_0}) = M(\underline{k_0}) + 1$



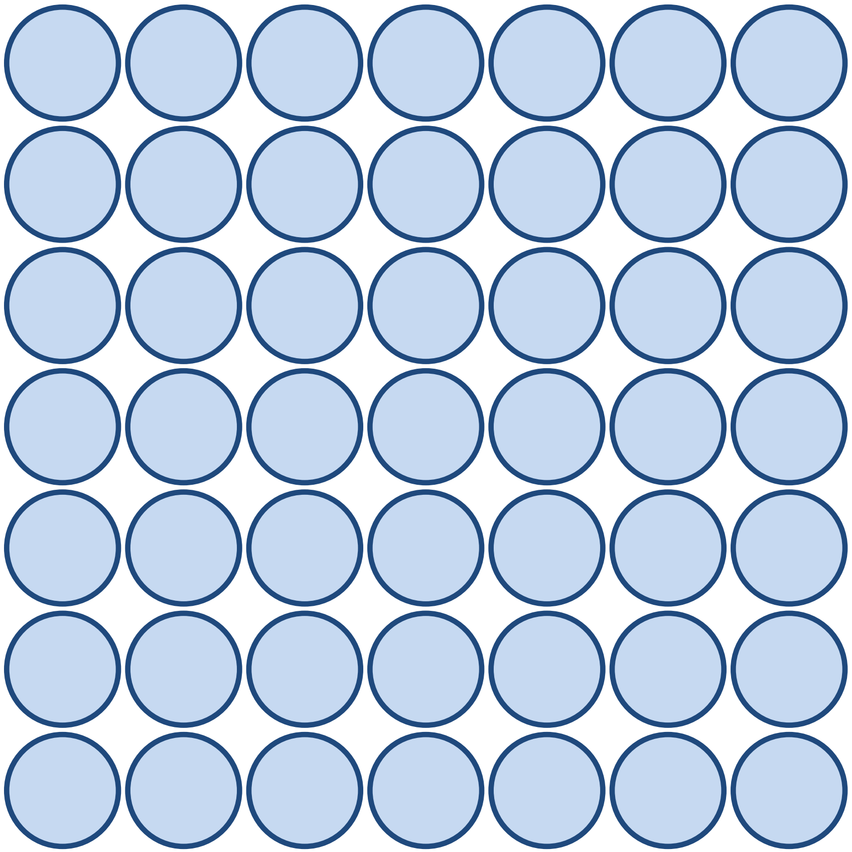
One neighbor k_0 and one neighbor k_1 :

$M(\underline{k_0}) = M(\underline{k_0}) + M(\underline{k_1}) + 1$

J. Hoshen and R. Kopelman. *Phys. Rev. B* **14**, 3438 (1976)

Algorithms

Newman and Ziff (microcanonical)

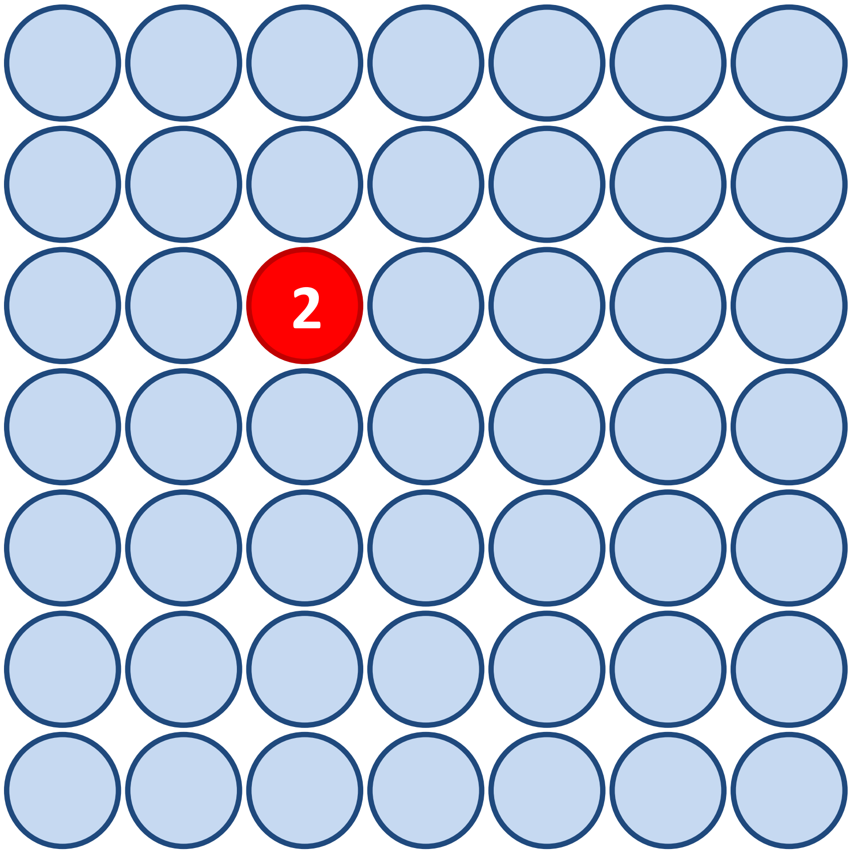


k	M(k)
2	0

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms

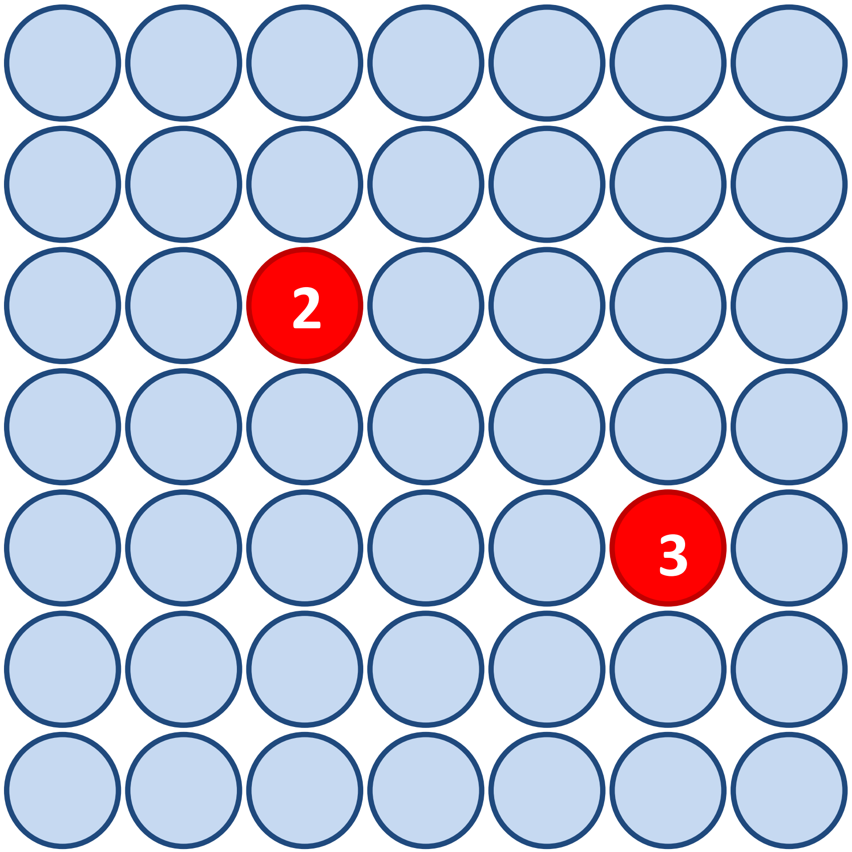
Newman and Ziff (microcanonical)



k	M(k)
2	1

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms
Newman and Ziff (microcanonical)

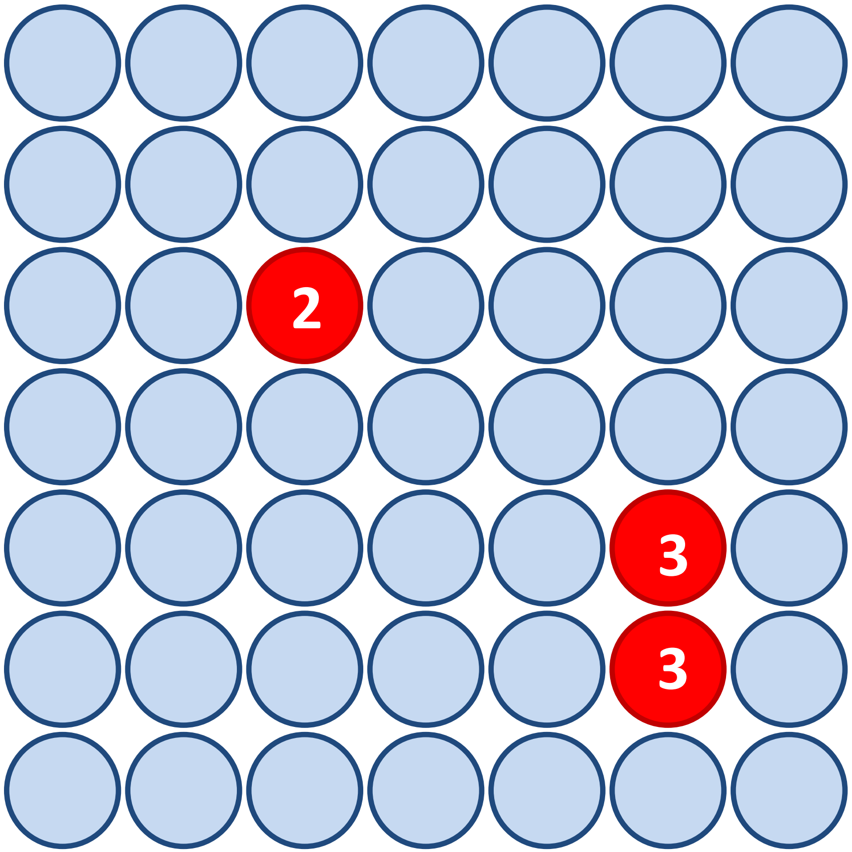


k	M(k)
2	1
3	1

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms

Newman and Ziff (microcanonical)

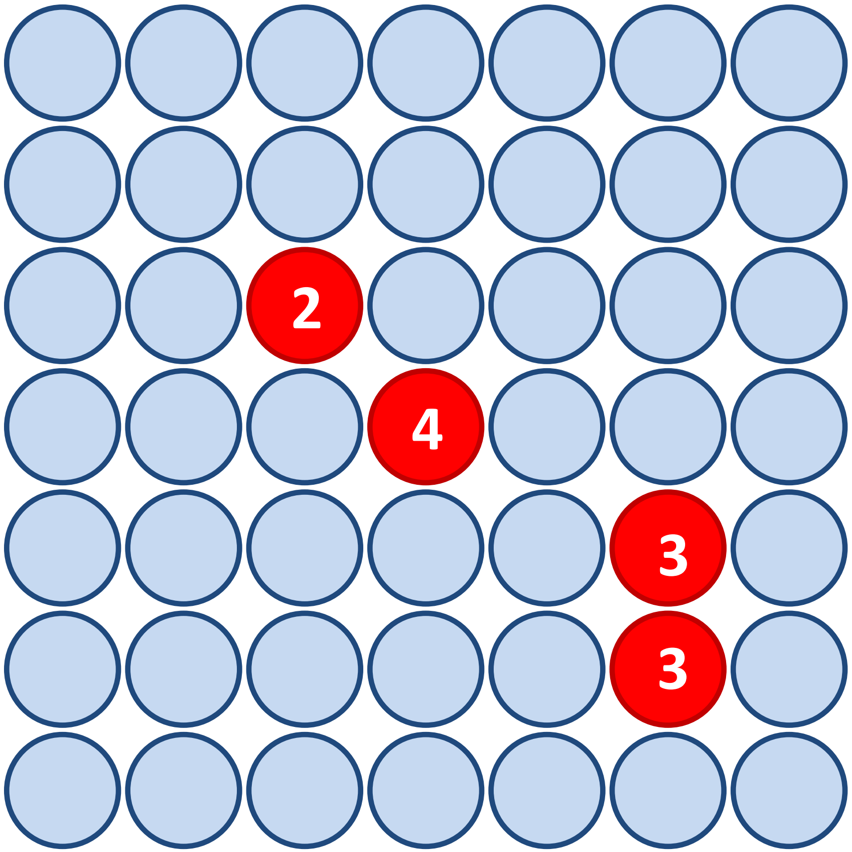


k	M(k)
2	1
3	2

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms

Newman and Ziff (microcanonical)

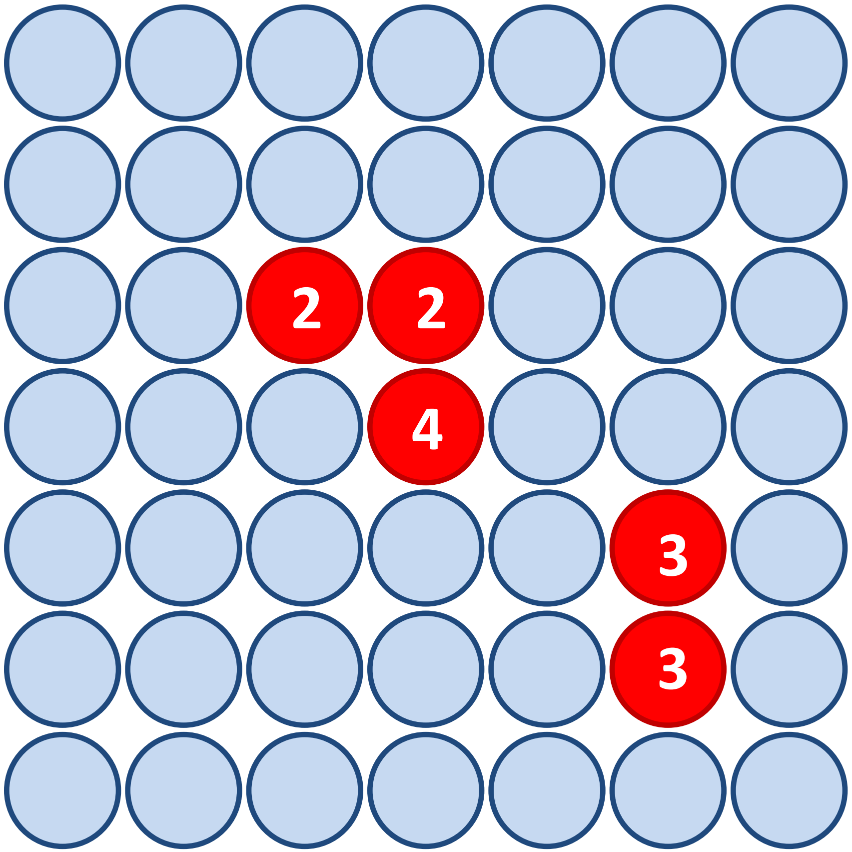


k	M(k)
2	1
3	2
4	1

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms

Newman and Ziff (microcanonical)



k	M(k)
2	3
3	2
4	-2

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)
M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)

Algorithms

Microcanonical vs canonical

Fixed number of occupied sites (*n*)

Fixed probability that a site is occupied (*p*)

$$\boldsymbol{B(N,n,p)} = \binom{N}{n} p^n (1-p)^{N-n}$$

$\boldsymbol{B(N,n,p)}$: probability that exactly n sites are occupied in a *canonical* configuration

$$Q(p) = \sum_{n=0}^{N} \boldsymbol{B(N,n,p)} Q_n = \sum_{n=0}^{N} \binom{N}{n} p^n (1-p)^{N-n} Q_n$$

M. E. J. Newman and R. M. Ziff. *Phys. Rev. Lett.* **85**, 4104 (2000)

M. E. J. Newman and R. M. Ziff. *Phys. Rev. E* **64**, 016706 (2001)