CFTC Centro de Física Teórica e Computacional

Ciências ULisboa

Percolation

Nuno Araújo

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

http://www.namaraujo.net

Books on percolation

➢D. Stauffer and A. Aharony, *Introduction to percolation theory*. CRC Press (2000).

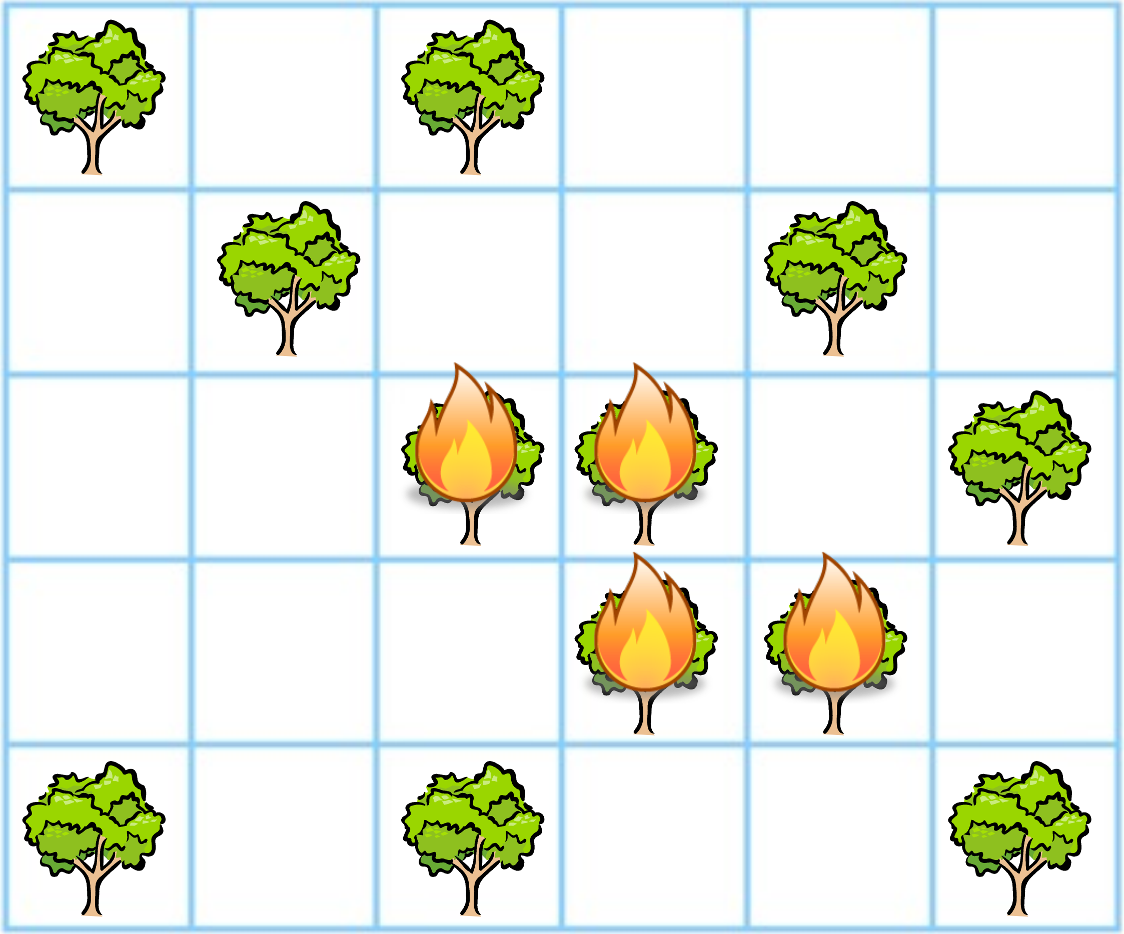
➢M. Sahimi, *Applications of percolation theory*. Taylor & Francis (1994).

➢K. Christensen and N. R. Moloney, *Complexity and criticality*. Imperial College Press (2005).

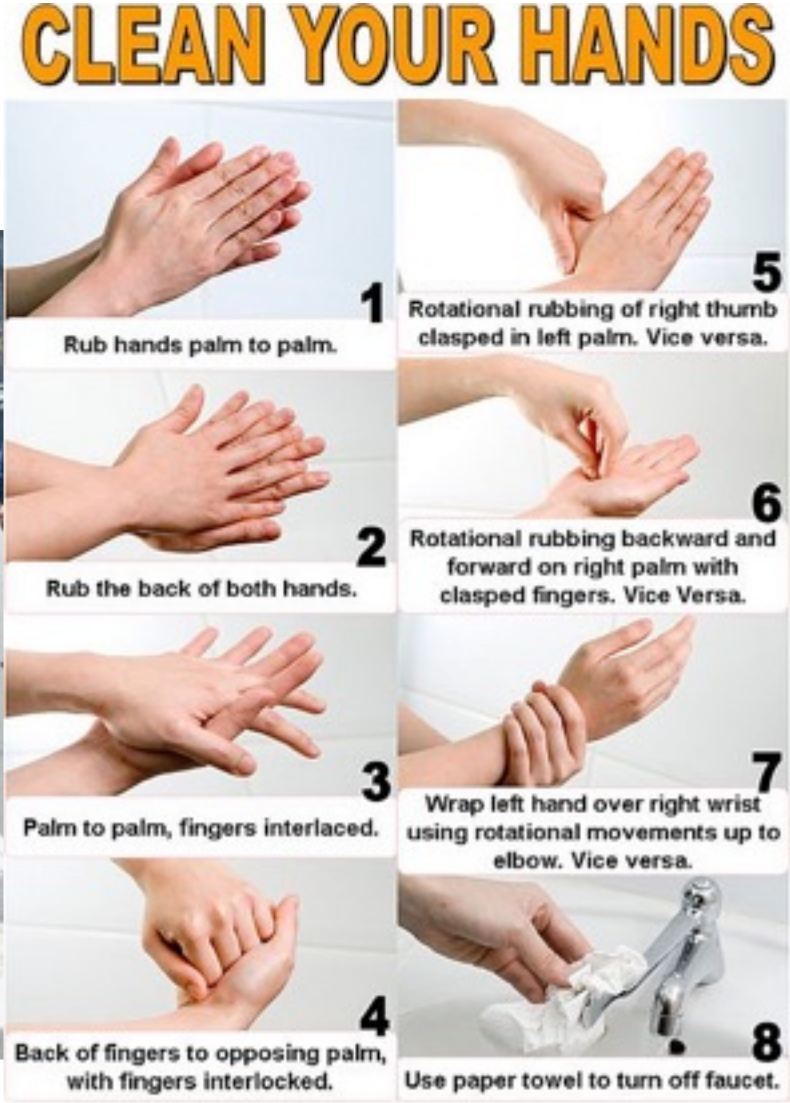
Forest fire

Photo - John McColgan BLM Alaska Fire Service

Forest fire



Spreading of epidemics



CLEAN YOUR HANDS

1 Rub hands palm to palm.

2 Rub the back of both hands.

3 Palm to palm, fingers interlaced.

4 Back of fingers to opposing palm, with fingers interlocked.

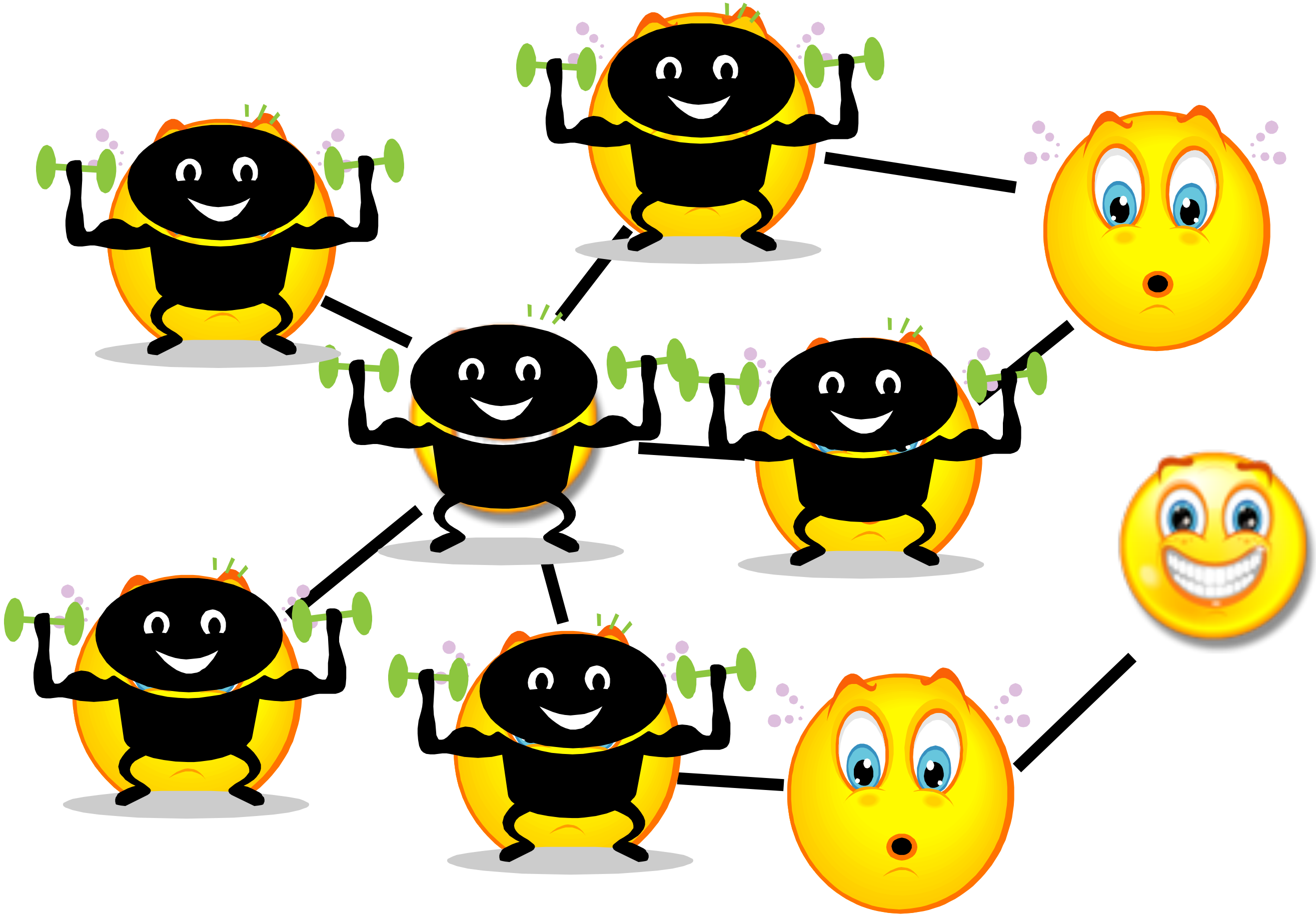
5 Rotational rubbing of right thumb clasped in left palm. Vice versa.

6 Rotational rubbing backward and forward on right palm with clasped fingers. Vice Versa.

7 Wrap left hand over right wrist using rotational movements up to elbow. Vice versa.

8 Use paper towel to turn off faucet.

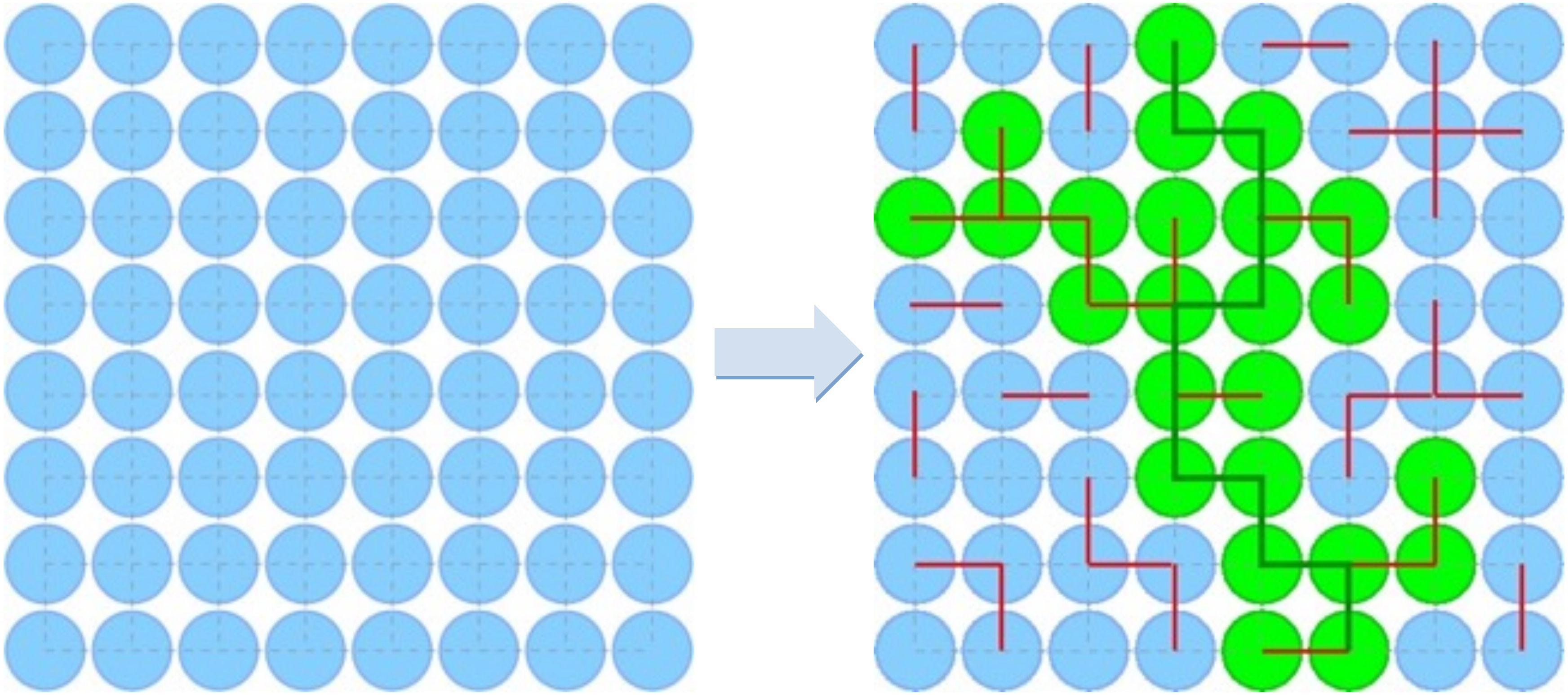
Spreading of epidemics



Oil fields

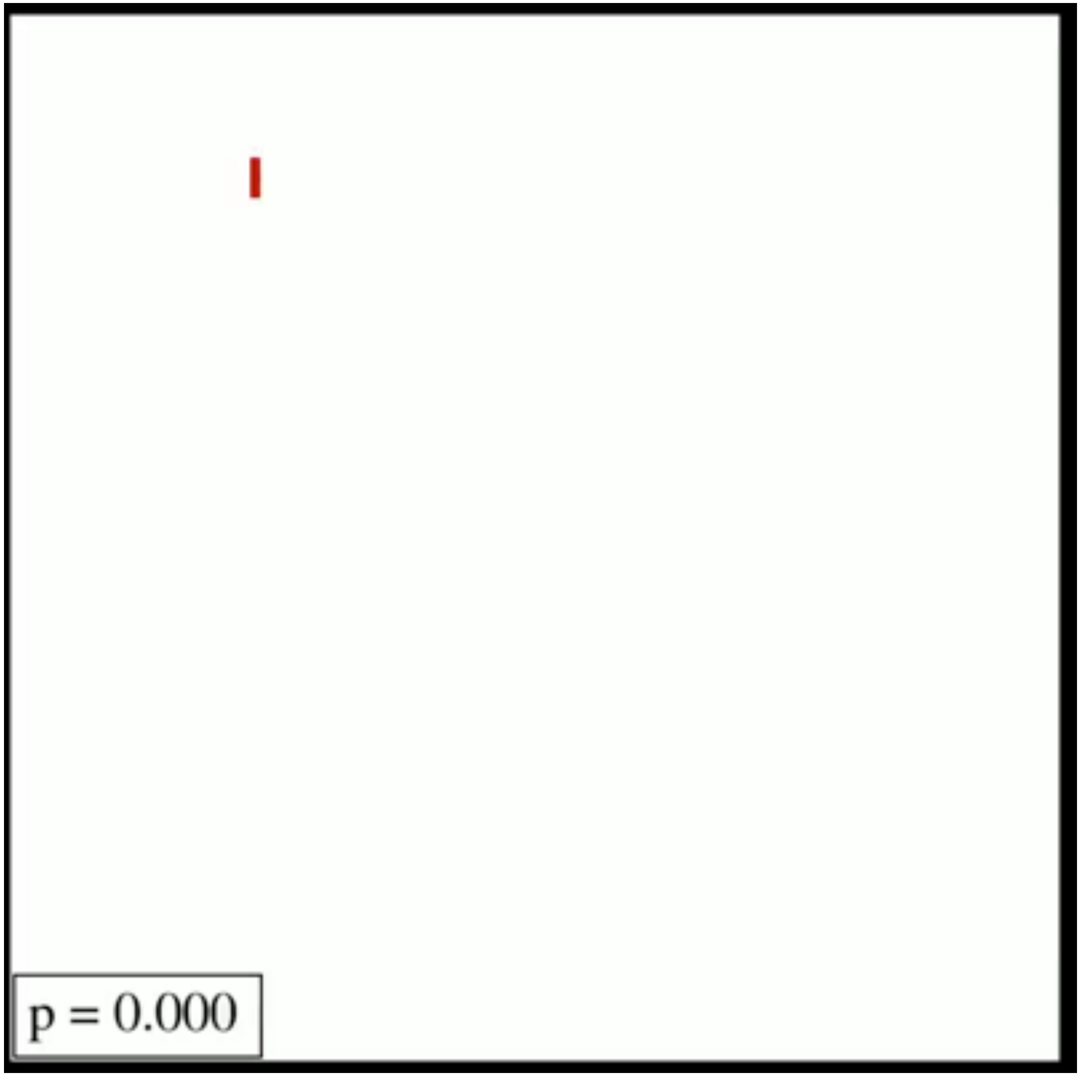
at *Barrancabermeja (Colombia)*, photo by *Melissa Jiménez*.

Percolation model



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

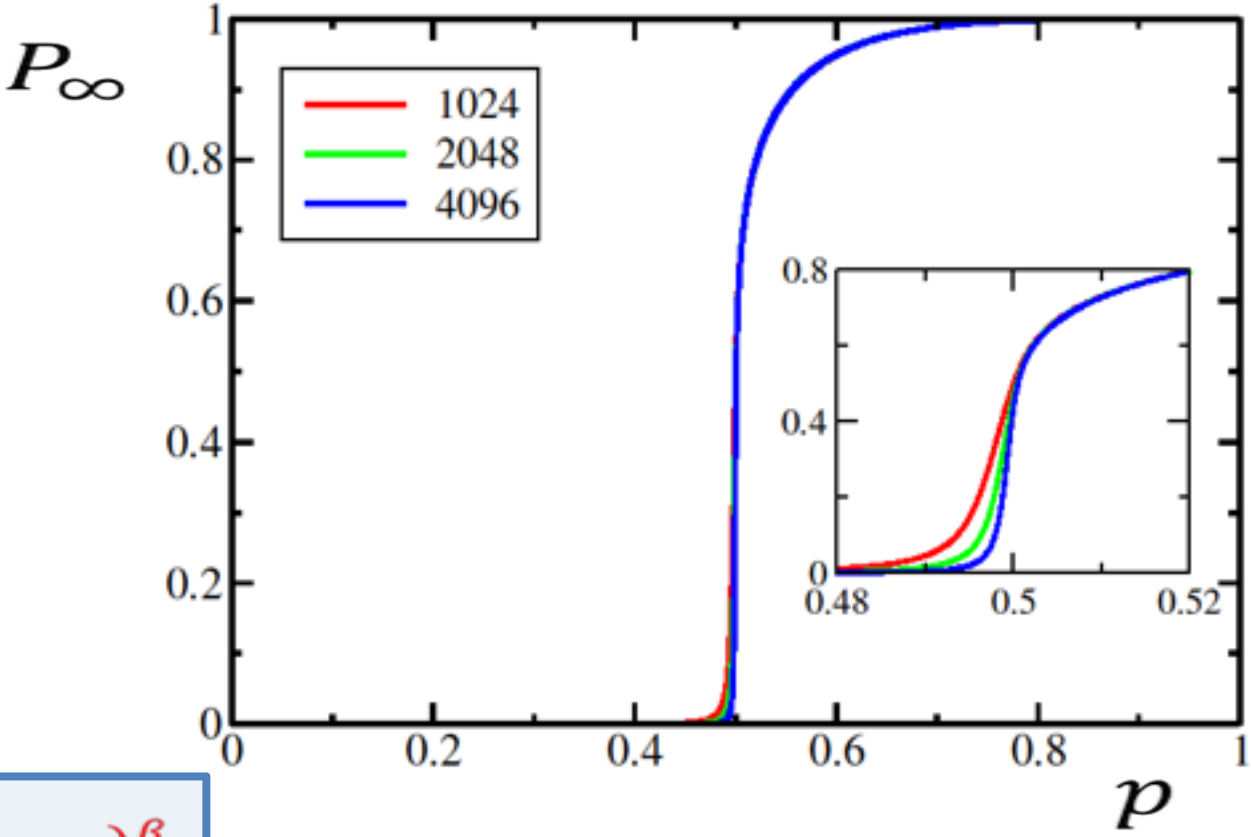
Percolation model



Percolation model

order parameter

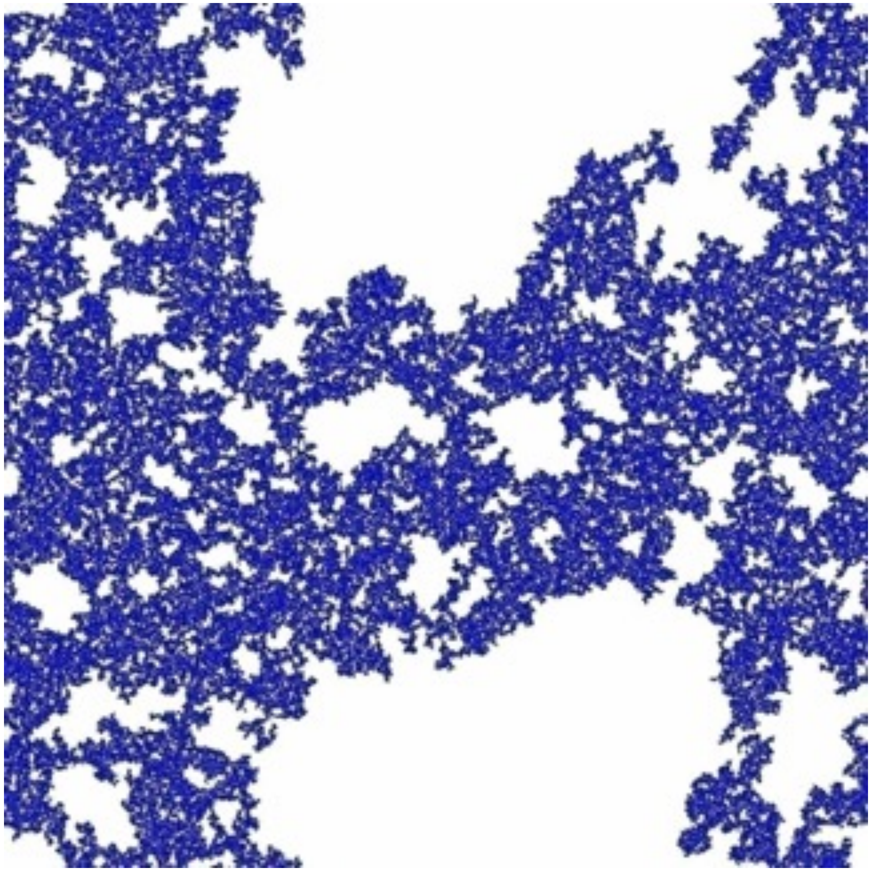
$$P_\infty = \frac{S_{max}}{N}$$



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

Percolation threshold

largest cluster: fractal dimension

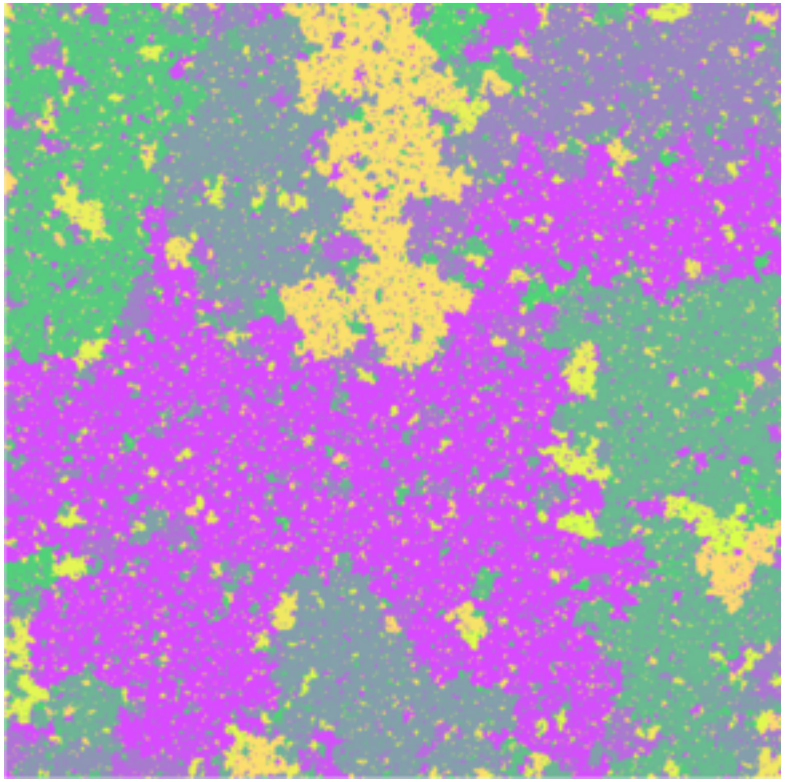


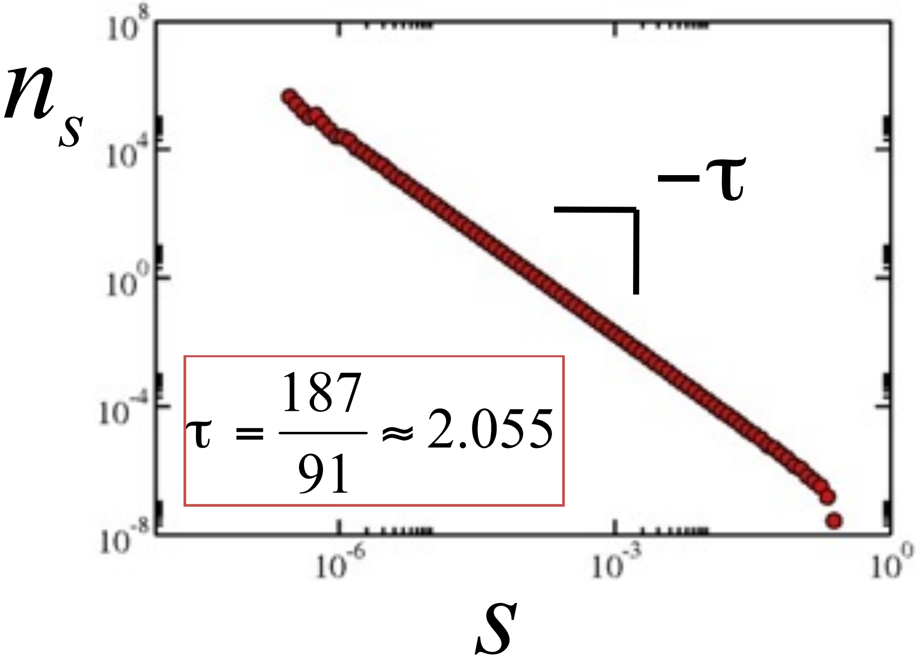
$$\boxed{d_f = \frac{91}{48} \approx 1.896}$$

Percolation threshold

cluster-size distribution

$$\boxed{n_s \sim s^{-\tau}}$$

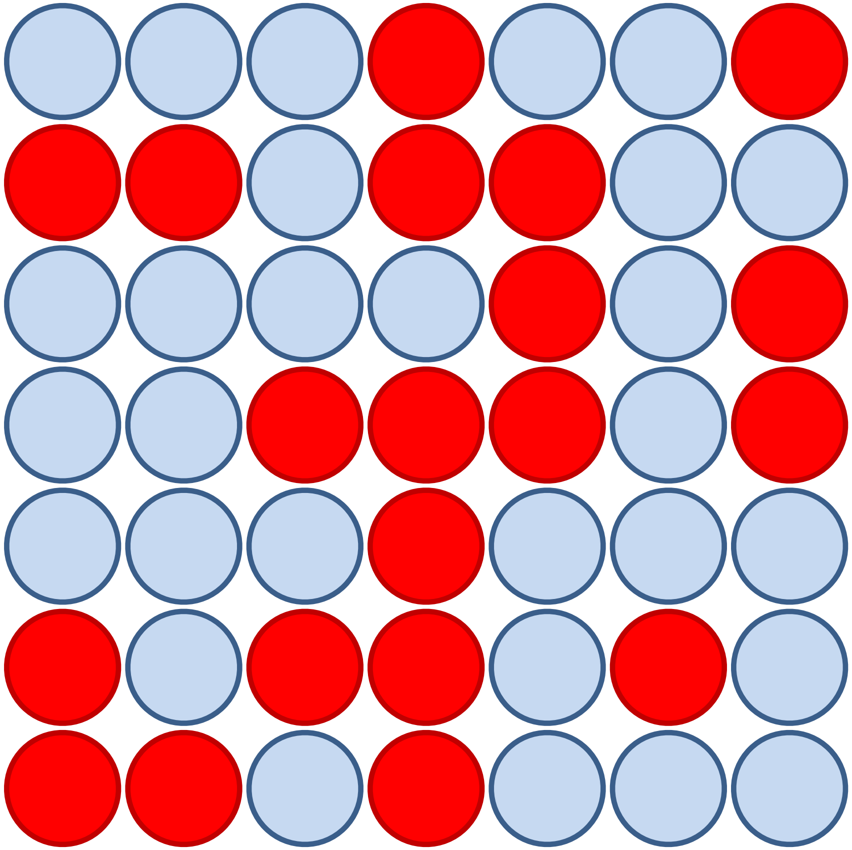




$$\boxed{\tau = \frac{187}{91} \approx 2.055}$$

Algorithms

generate canonical configurations



```
For each site i:
1. random number $\varepsilon$;
2. if
      $\varepsilon < p$: i is occupied;
      else: i is empty.
```

Algorithms

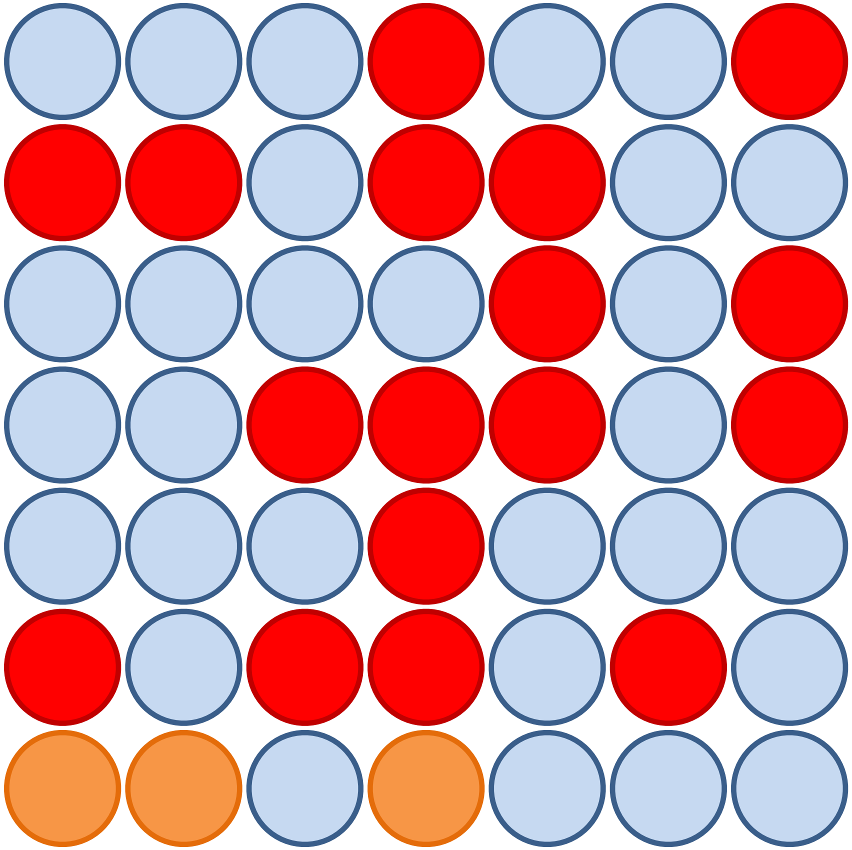
Burning method

🔵 empty

🔴 occupied

🟠 burning

⚫ burned



🔵	🔵	🔵	🔴	🔵	🔵	🔴
🔴	🔴	🔵	🔴	🔴	🔵	🔵
🔵	🔵	🔵	🔵	🔴	🔵	🔴
🔵	🔵	🔴	🔴	🔴	🔵	🔴
🔵	🔵	🔵	🔴	🔵	🔵	🔵
🔴	🔵	🔴	🔴	🔵	🔴	🔵
🟠	🟠	🔵	🟠	🔵	🔵	🔵

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

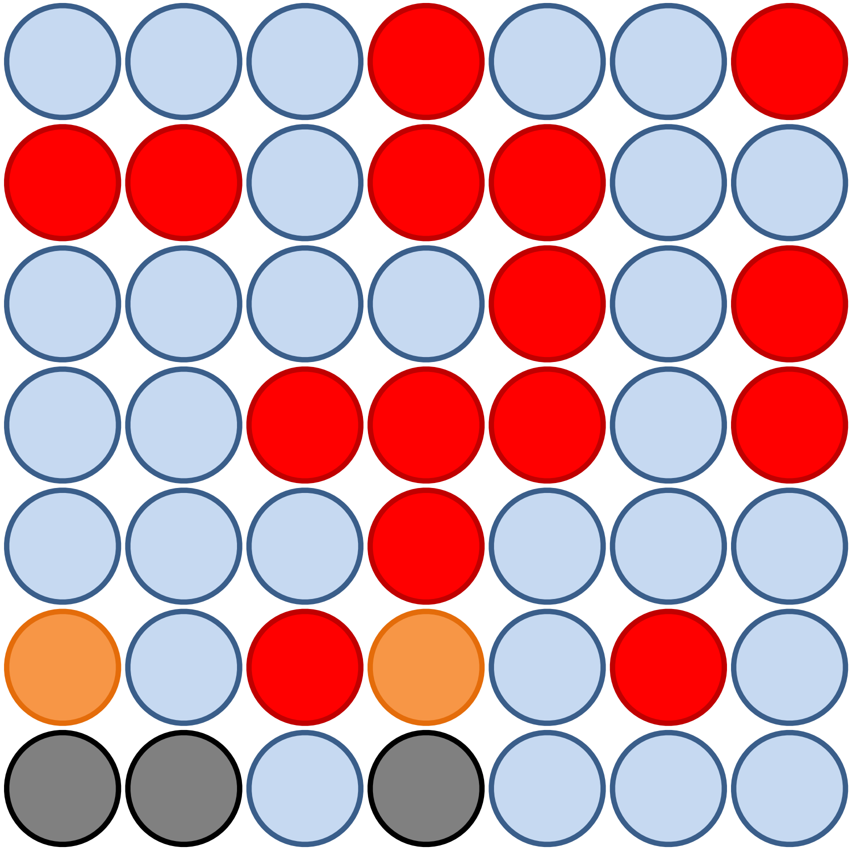
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ **empty** 🟠 **burning**

🔴 **occupied** ⚫ **burned**



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

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

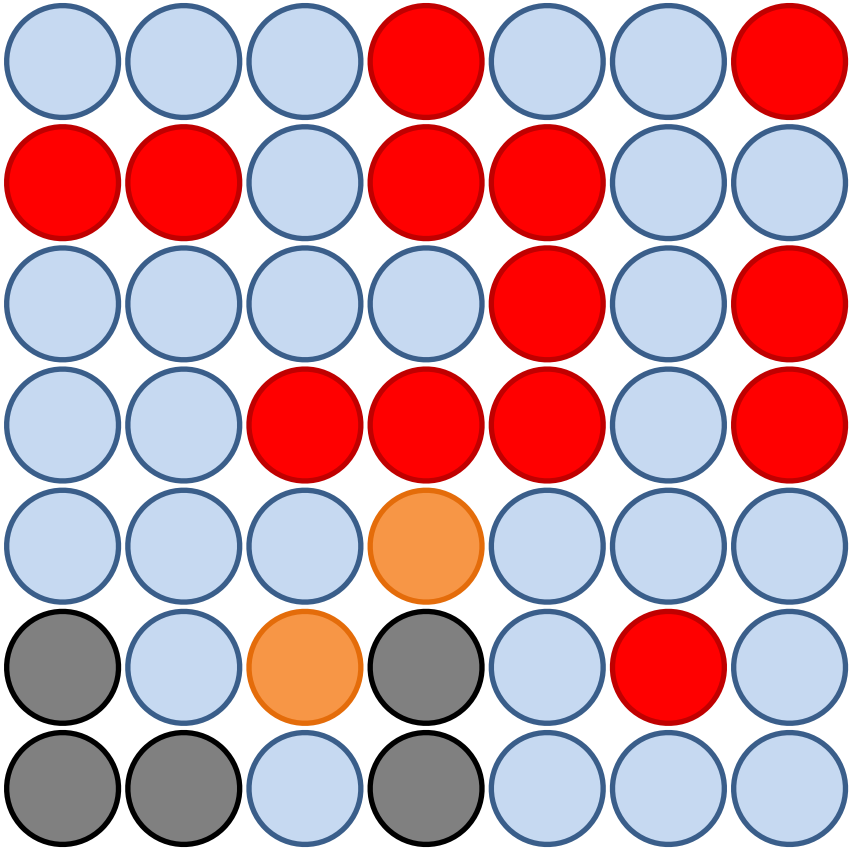
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ empty 🟠 burning

🔴 occupied ⚫ burned



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

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

3. **repeat** until everything is **burned**.

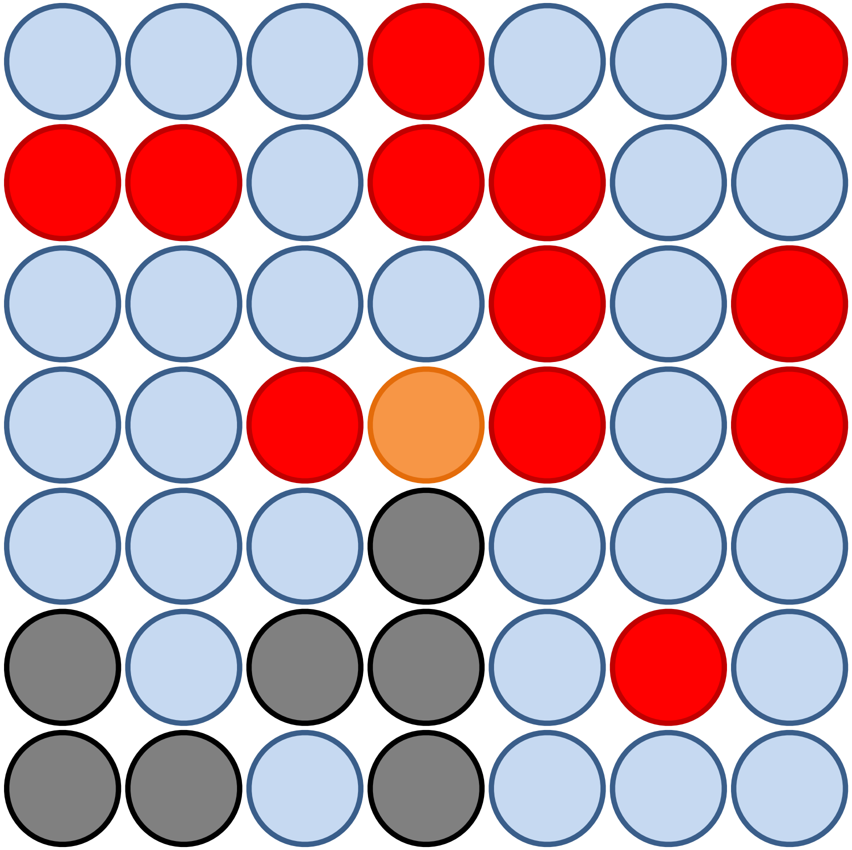
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

🔵 empty 🟠 burning

🔴 occupied ⚫ burned



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

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

3. **repeat** until everything is **burned**.

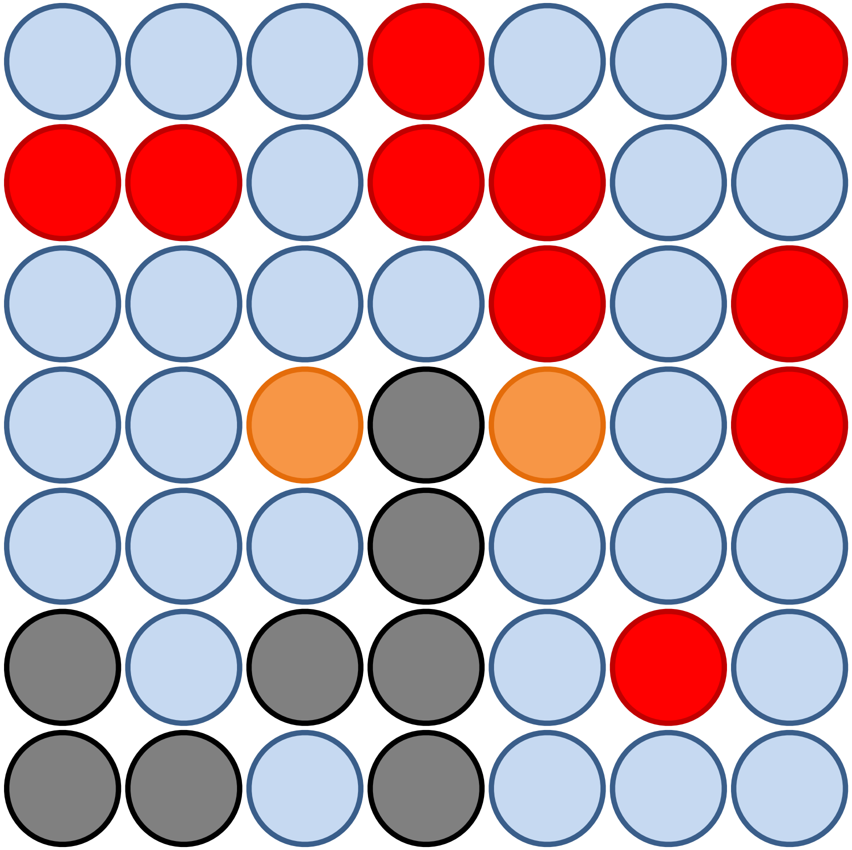
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ empty 🟠 burning

🔴 occupied ⚫ burned



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

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

3. **repeat** until everything is **burned**.

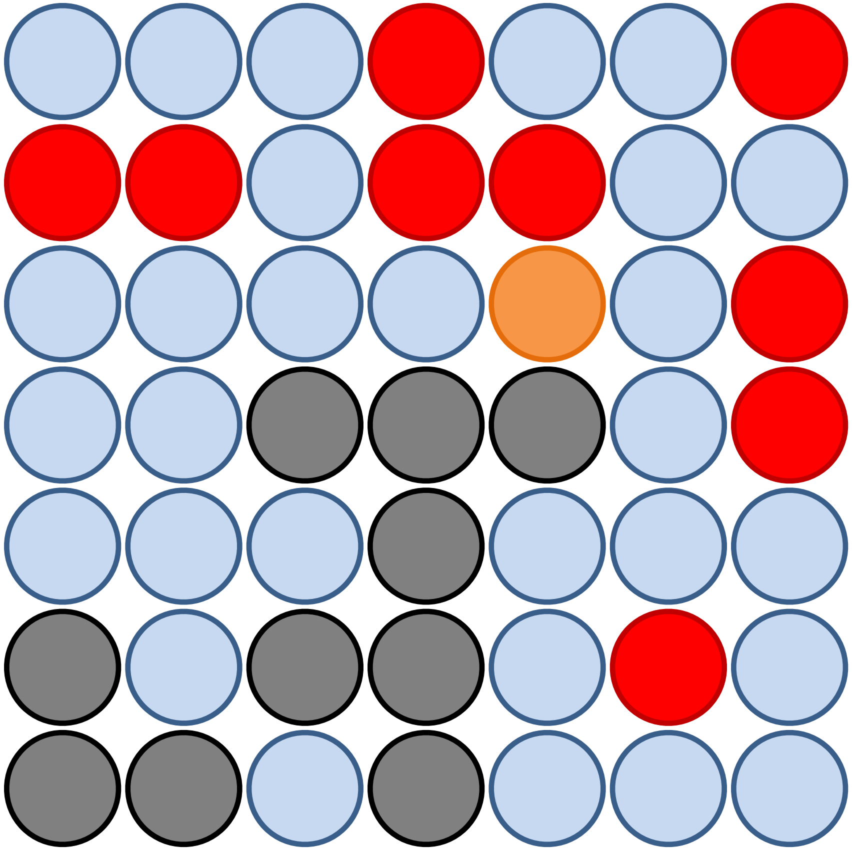
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ **empty** 🟠 **burning**

🔴 **occupied** ⚫ **burned**



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

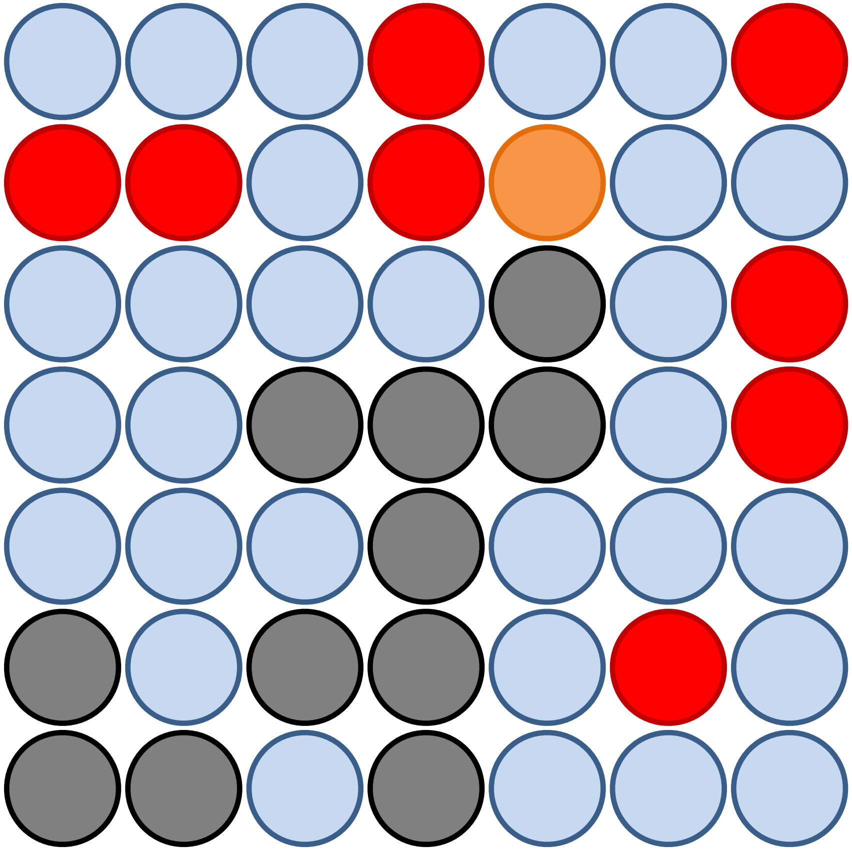
Burning method

🔵 **empty**

🟠 **burning**

🔴 **occupied**

⚫ **burned**



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

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

3. **repeat** until everything is **burned**.

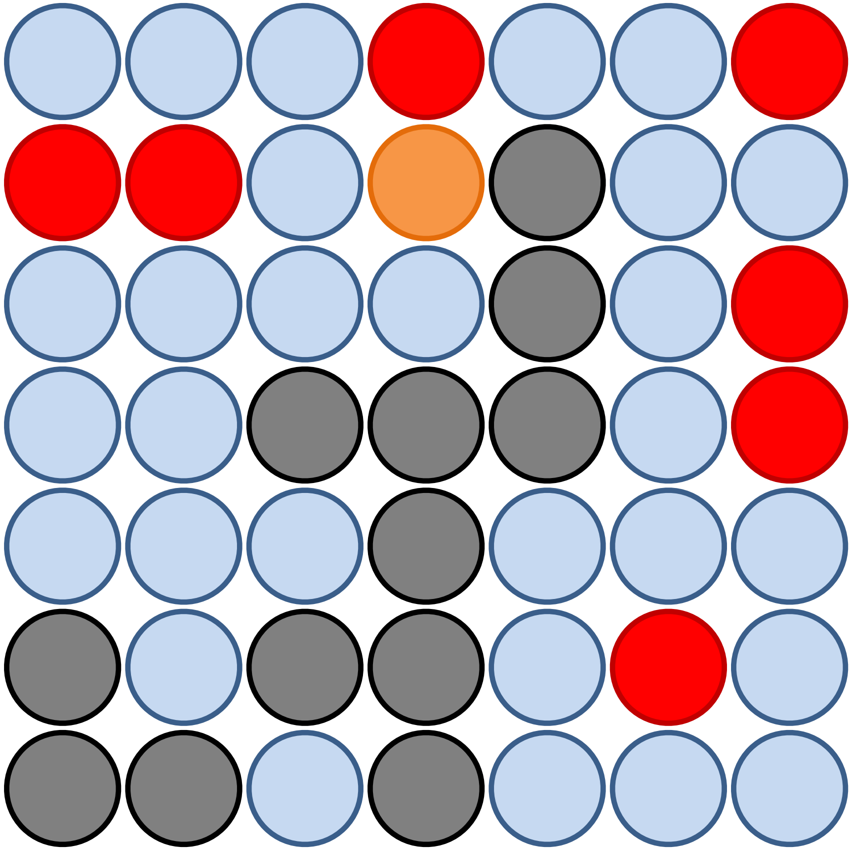
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ empty 🟠 burning

🔴 occupied ⚫ burned



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

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

3. **repeat** until everything is **burned**.

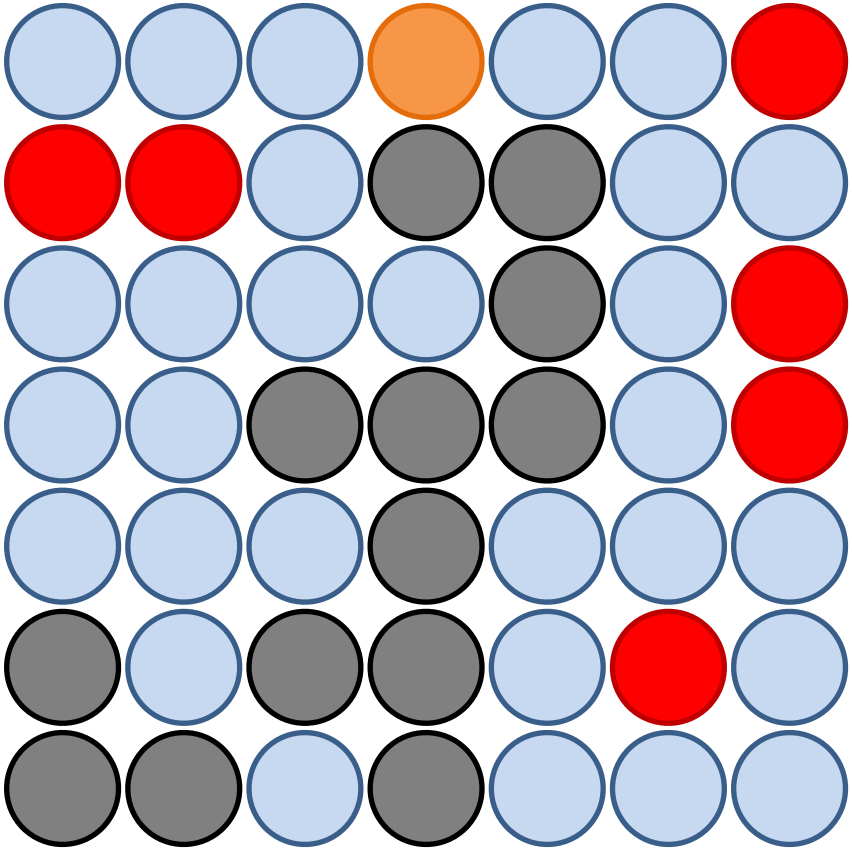
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

🔵 **empty** 🟠 **burning**

🔴 **occupied** ⚫ **burned**



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

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

3. **repeat** until everything is **burned**.

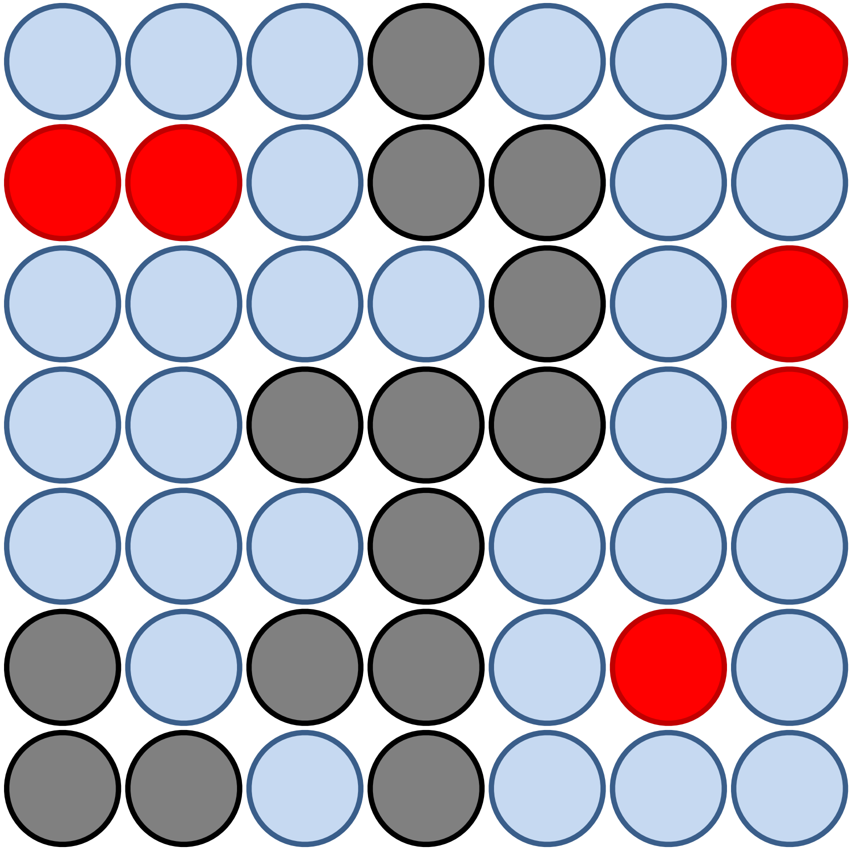
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

🔵 **empty** 🟠 **burning**

🔴 **occupied** ⚫ **burned**



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

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

3. **repeat** until everything is **burned**.

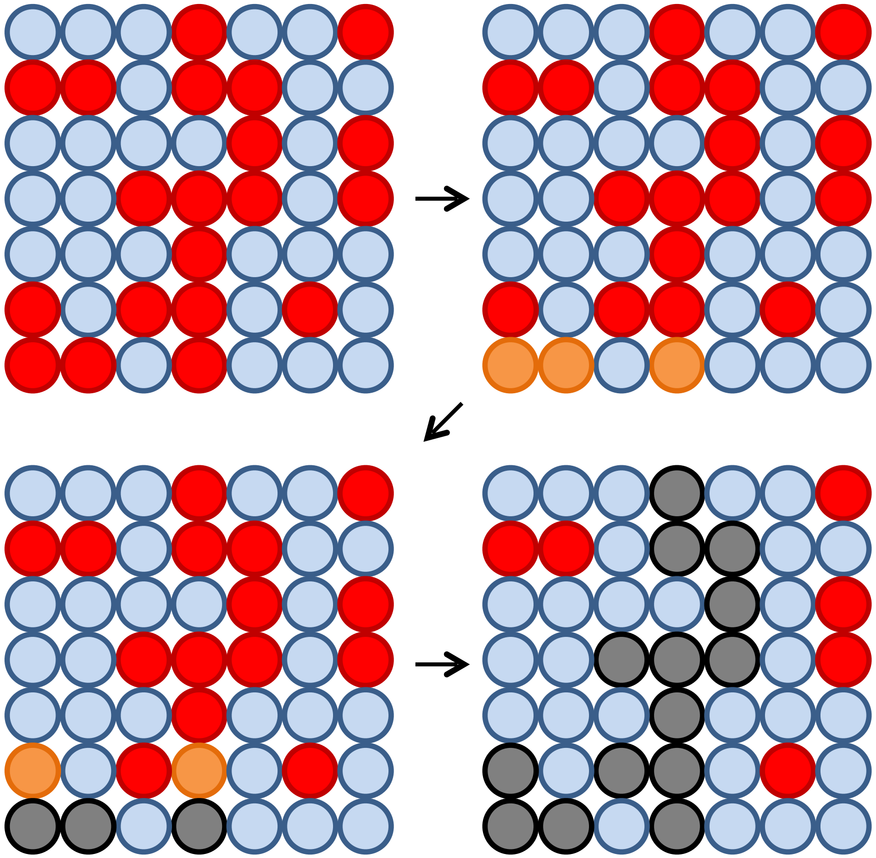
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

Algorithms

Burning method

○ empty 🟠 burning

🔴 occupied ⚫ burned



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

Number of clusters and cluster size distribution?

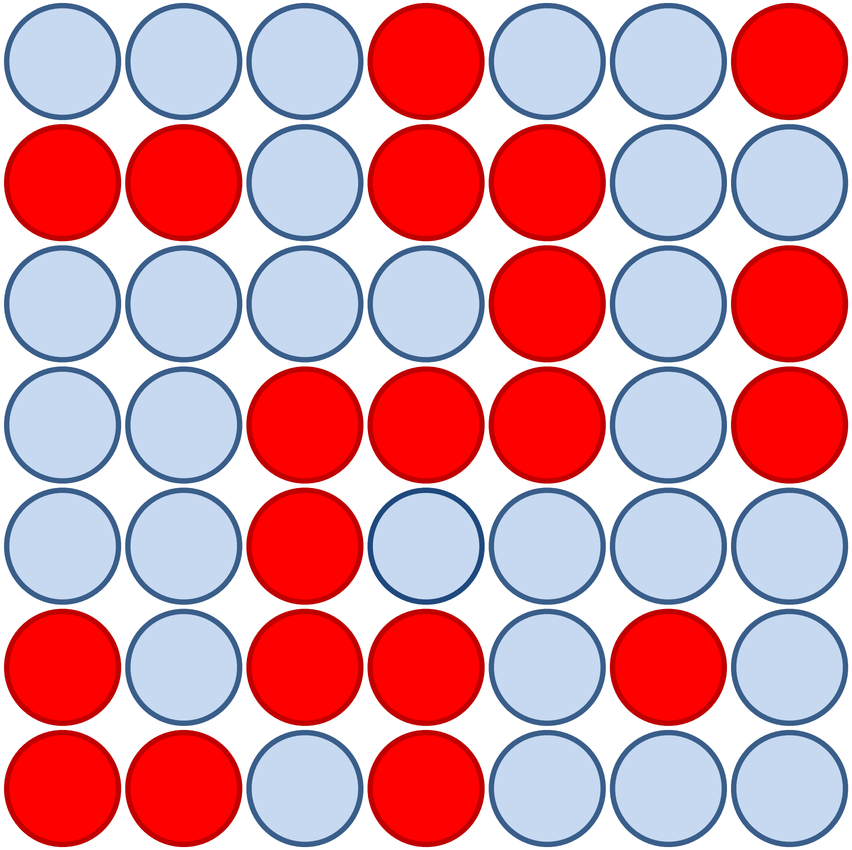
H. J. Herrmann, D. C. Hong, and H. E. Stanley. J. *Phys. A* **17**, L261 (1984)

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)

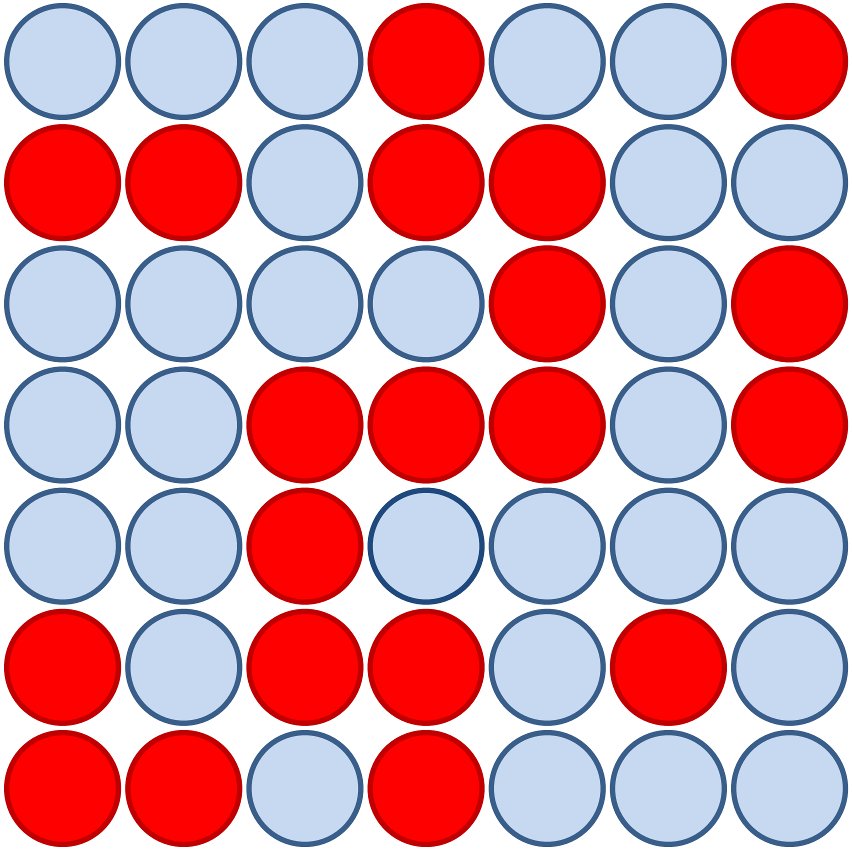
Algorithms

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)

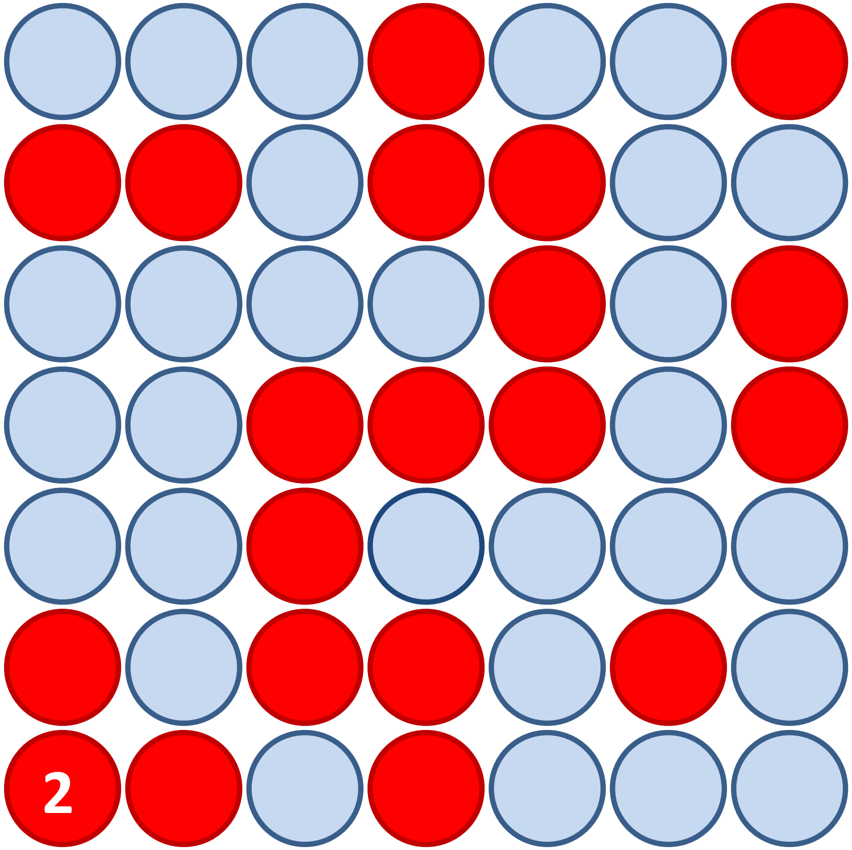
Algorithms

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

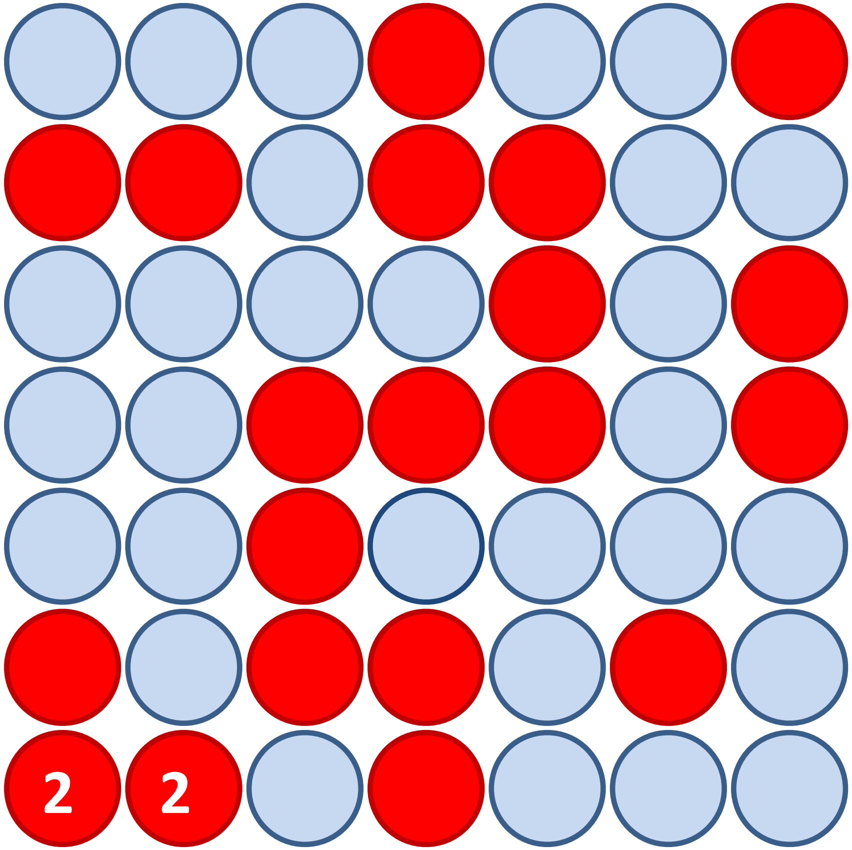
Hoshen and Kopelman

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)

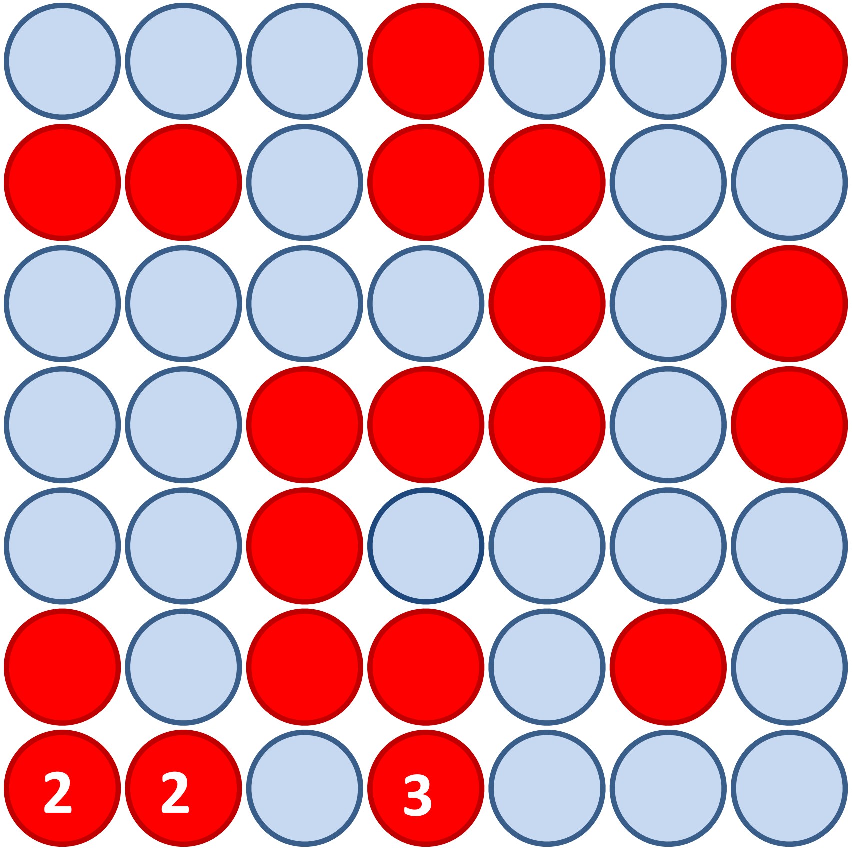
Algorithms

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.

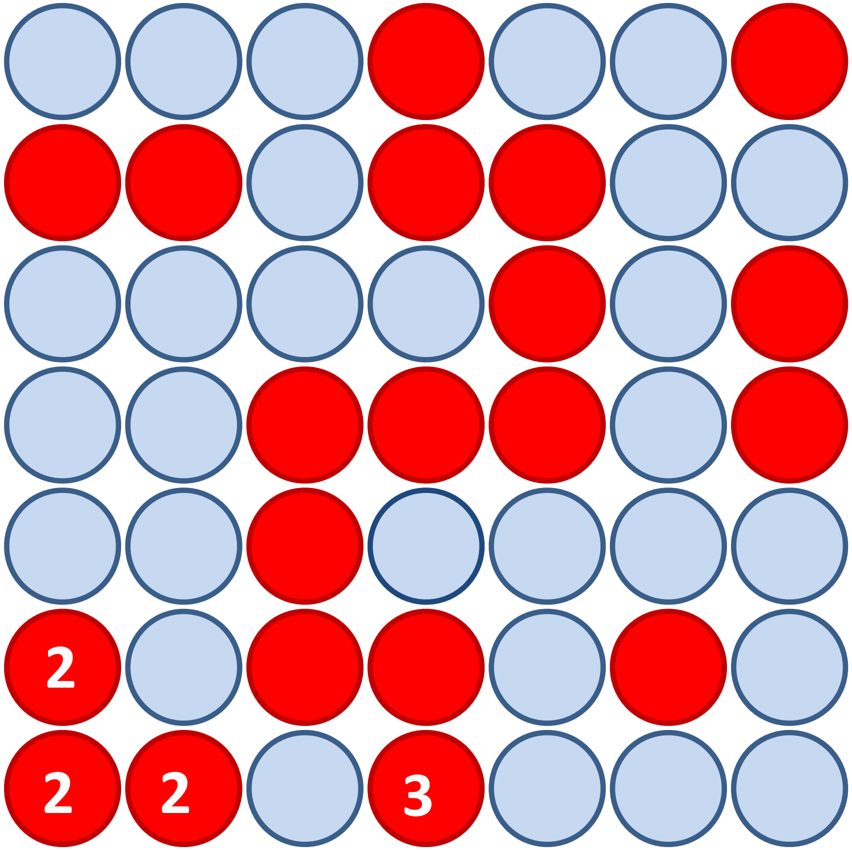


k	M(k)
2	2
3	1

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)

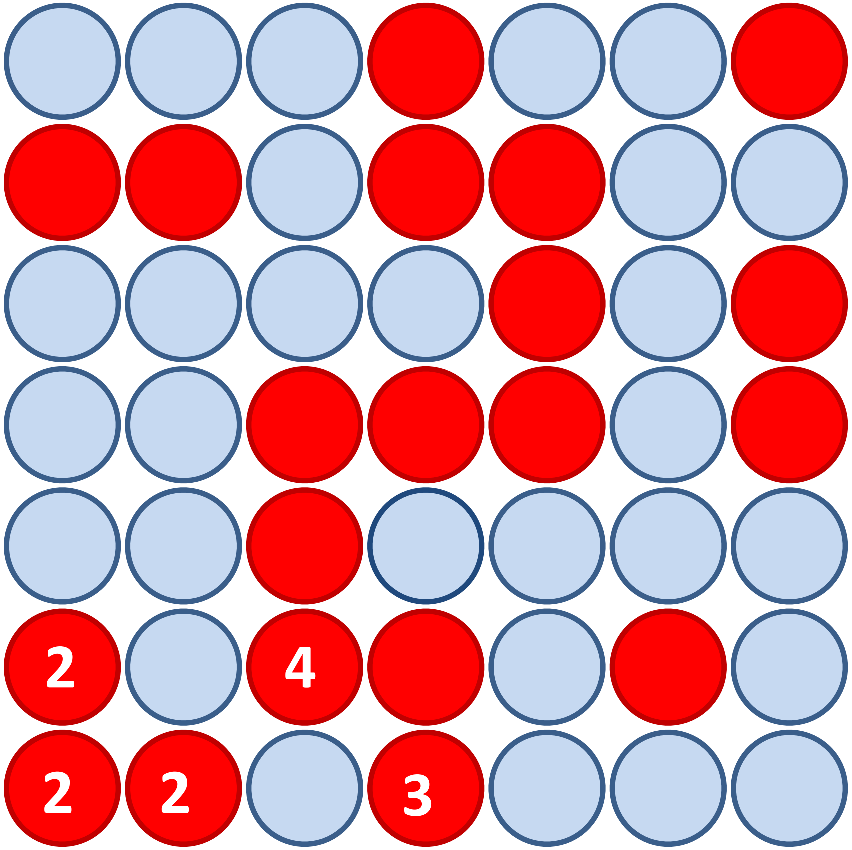
Algorithms

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)

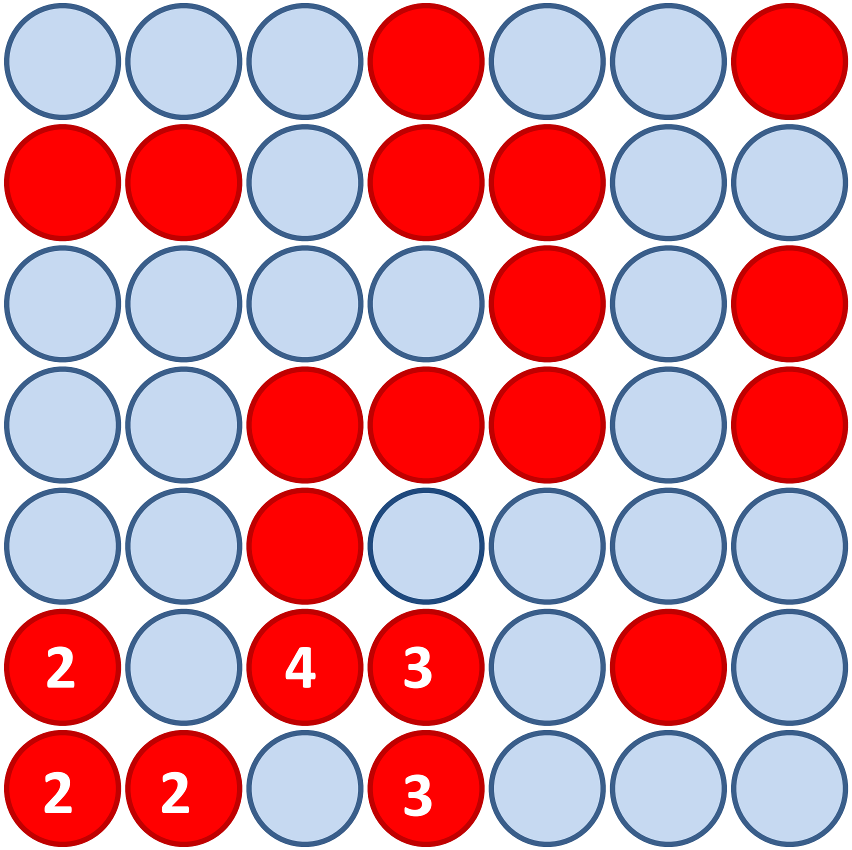
Algorithms

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)

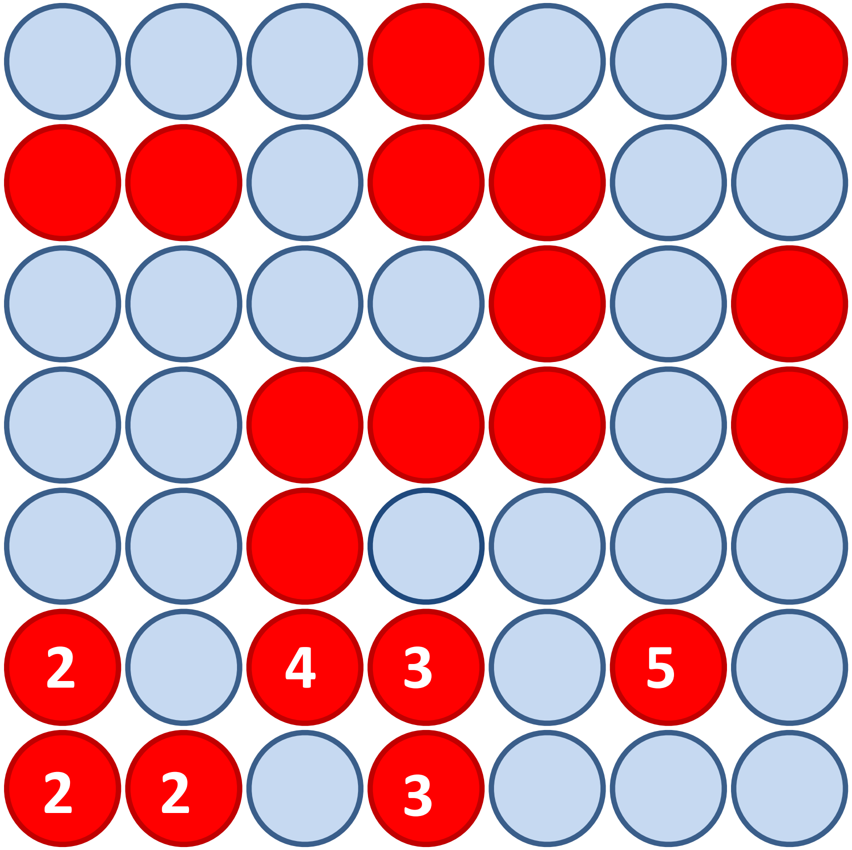
Algorithms

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)

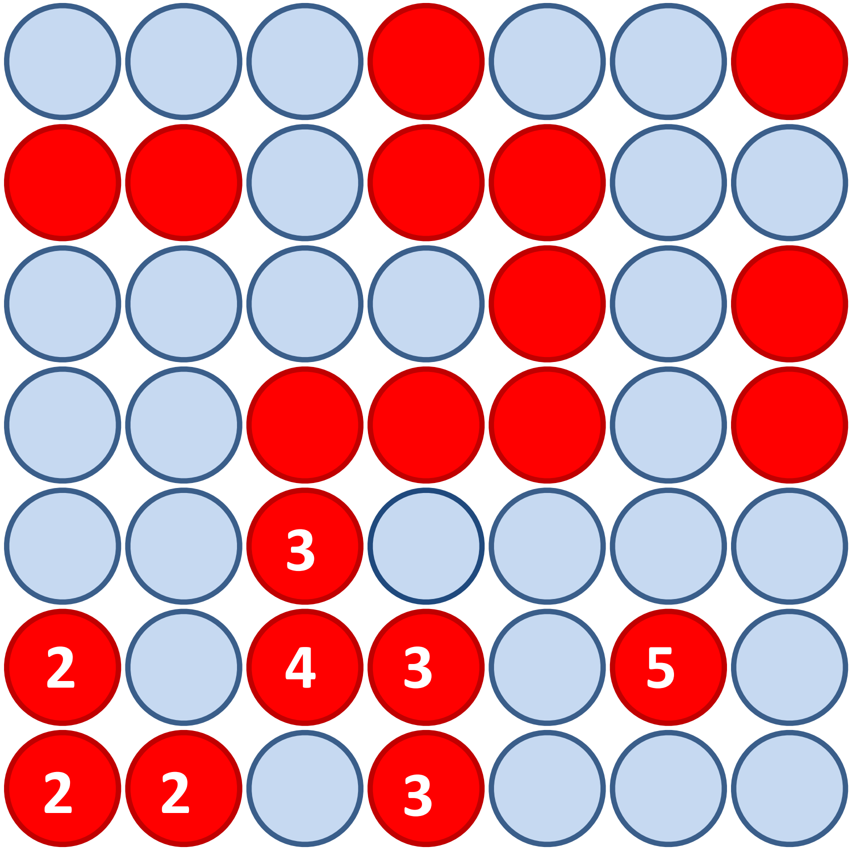
Algorithms

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)

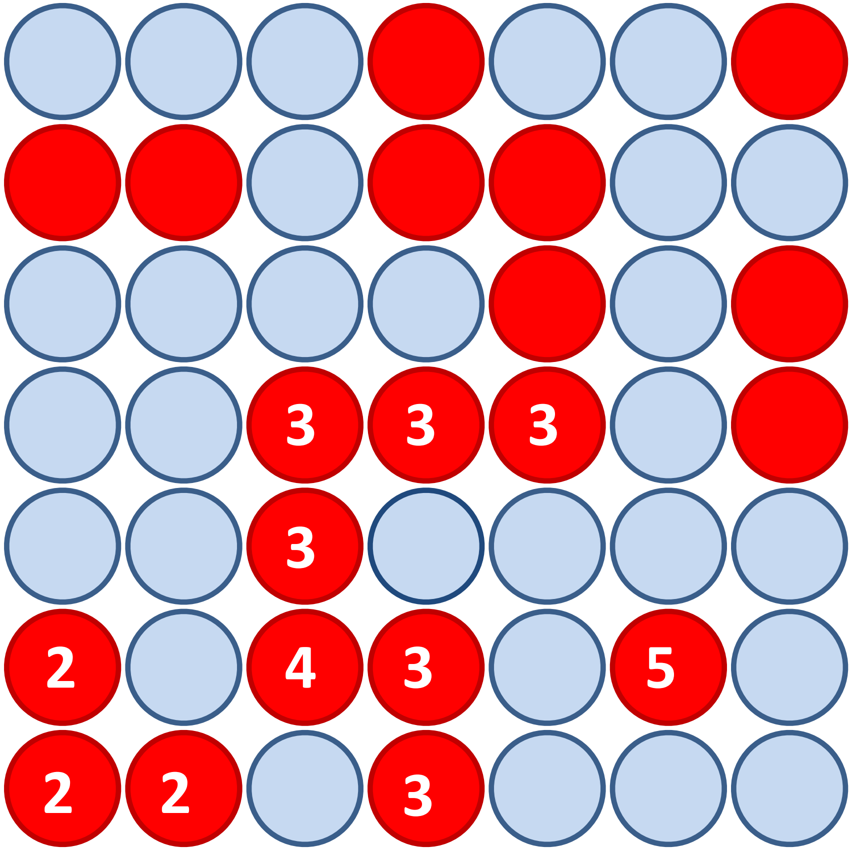
Algorithms

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)

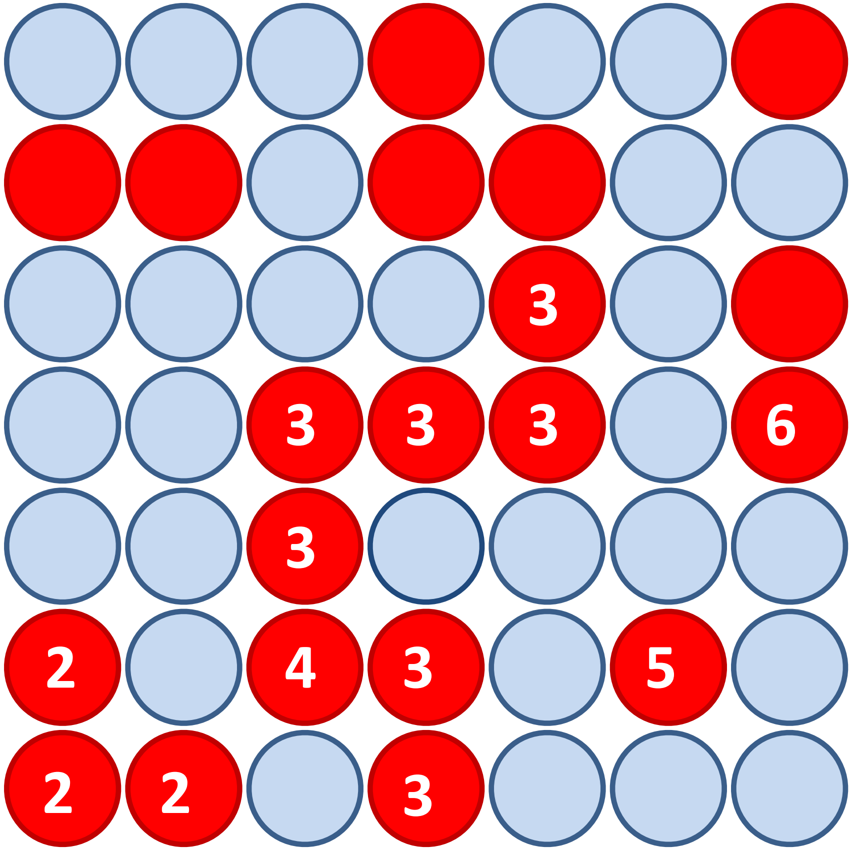
Algorithms

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)

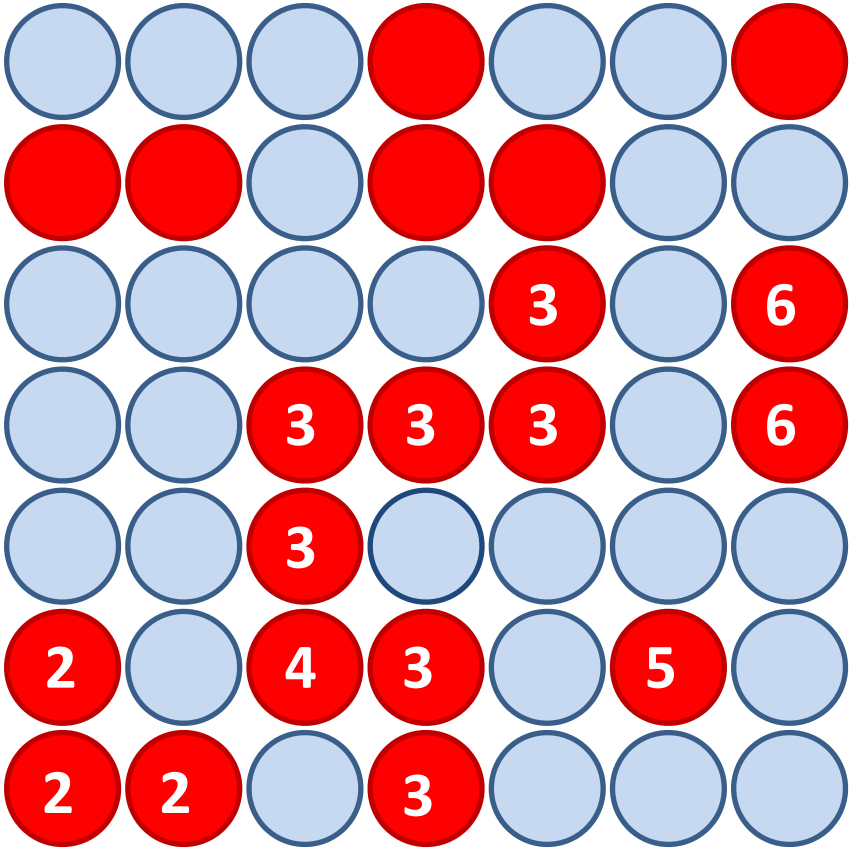
Algorithms

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)

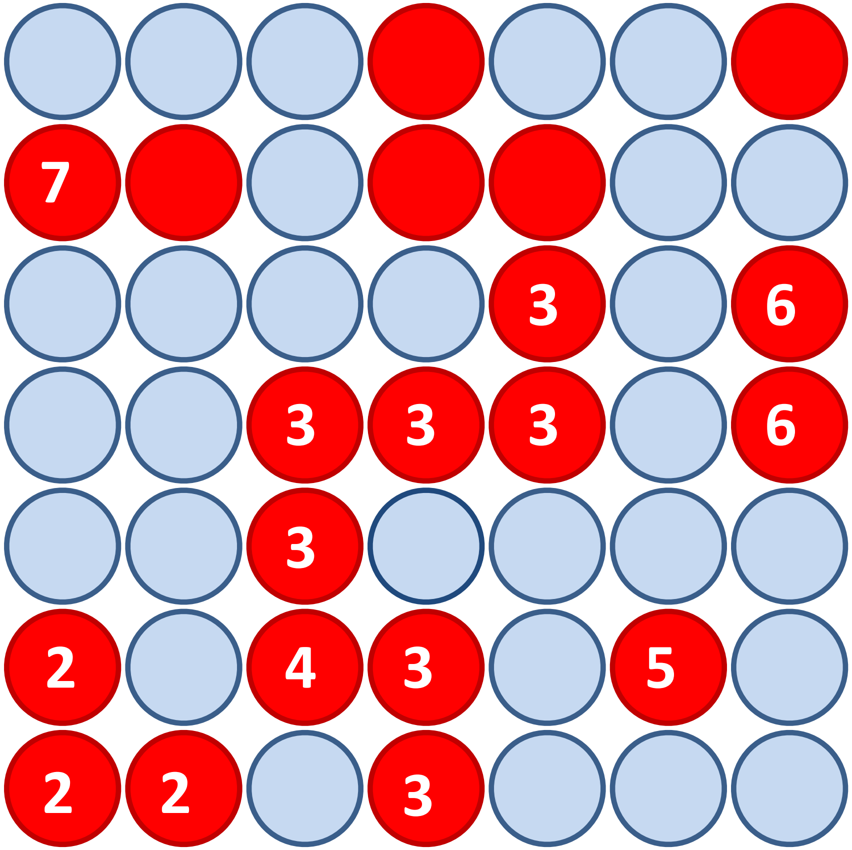
Algorithms

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)

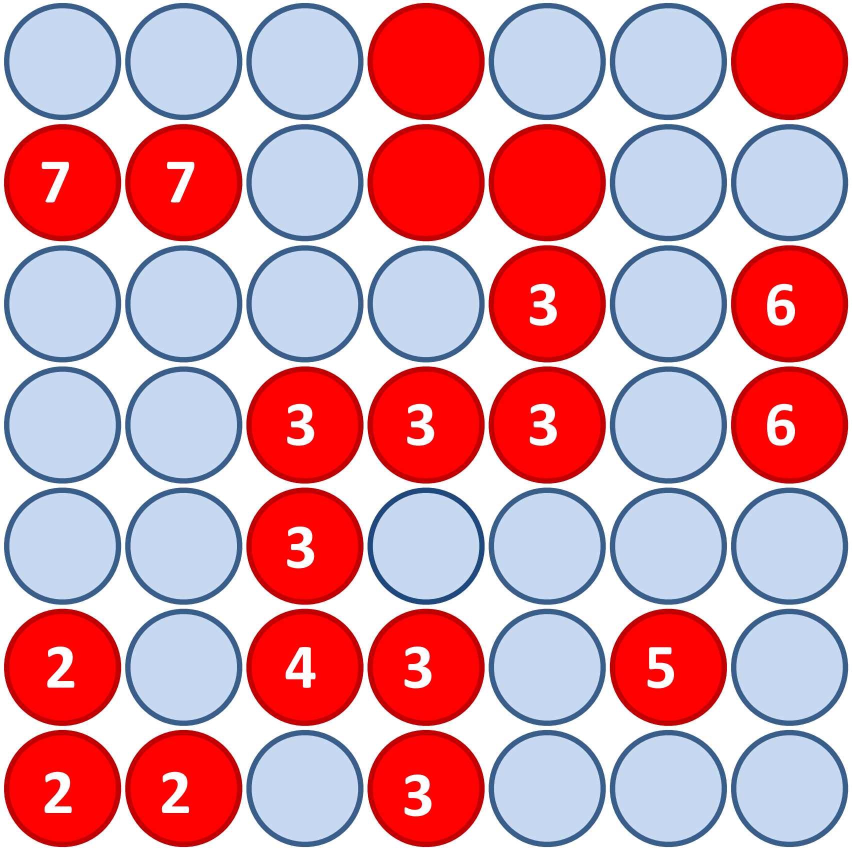
Algorithms

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)

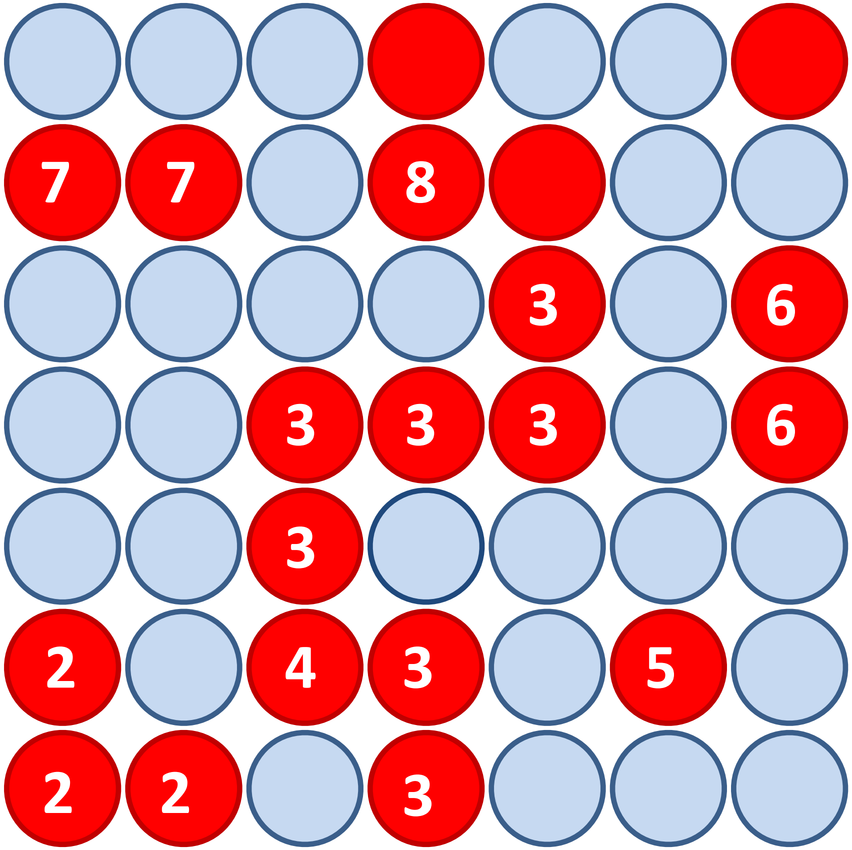
Algorithms

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)

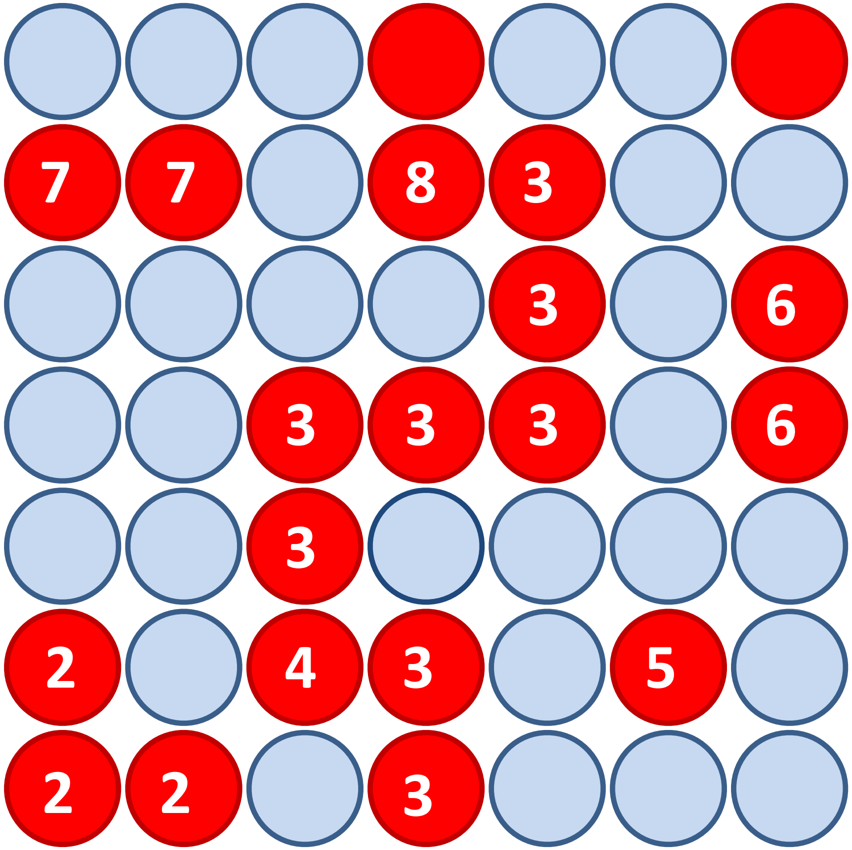
Algorithms

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)

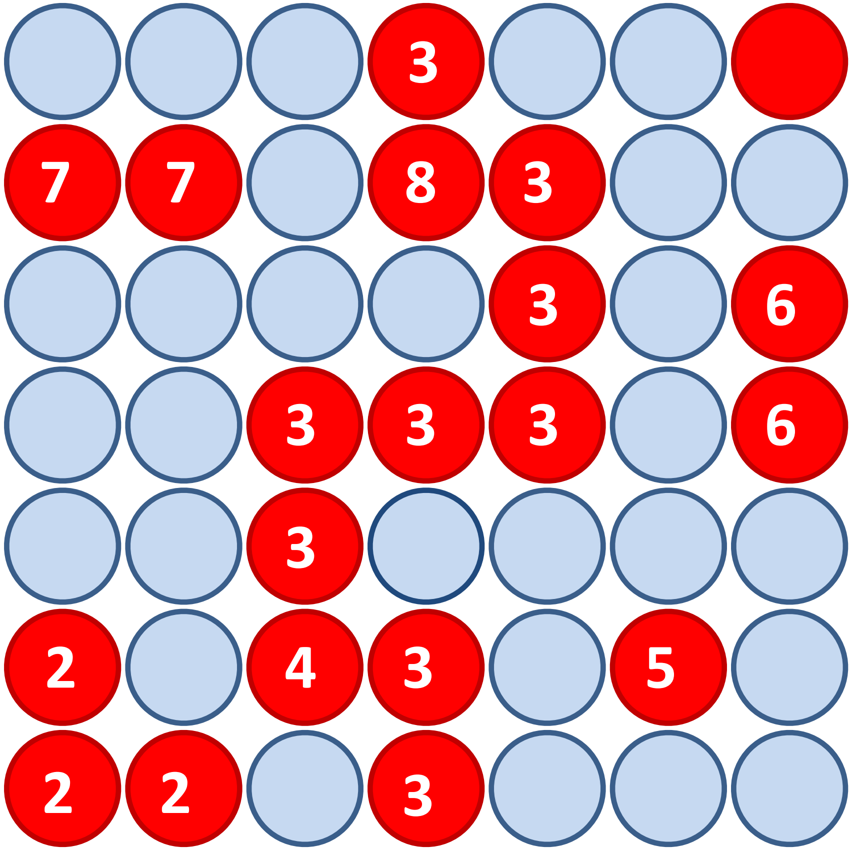
Algorithms

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)

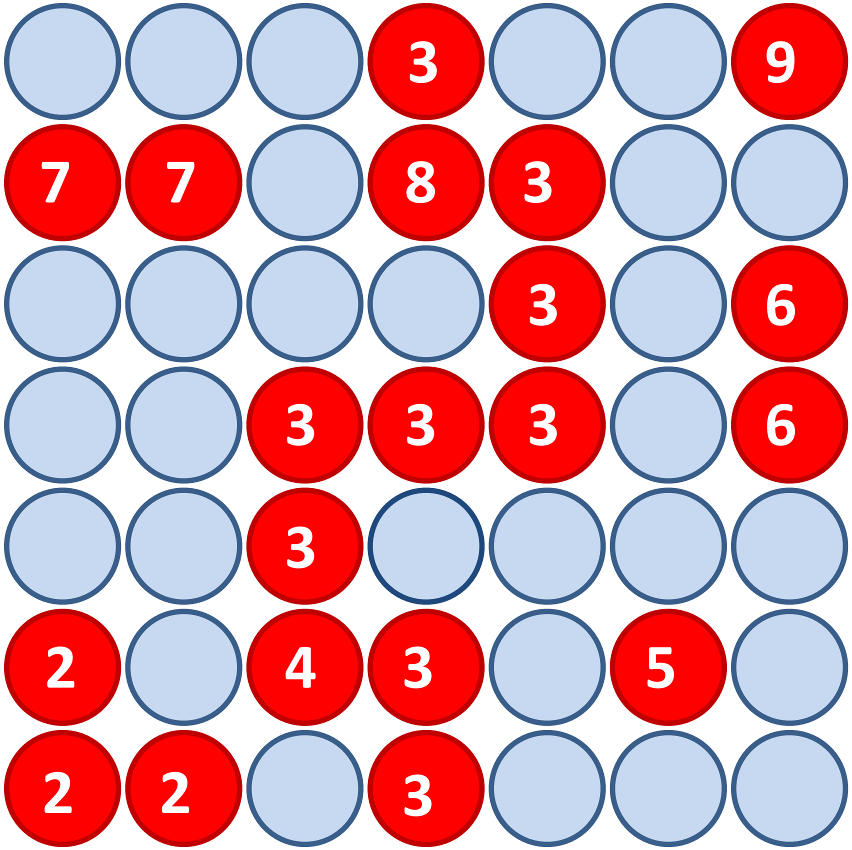
Algorithms

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.

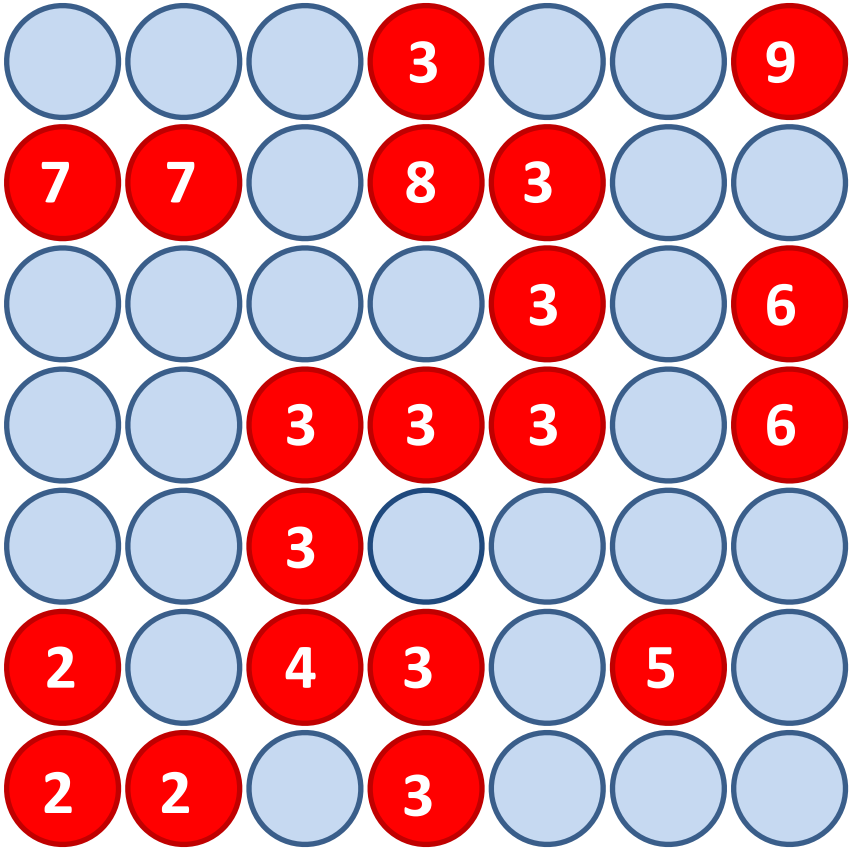


k	M(k)
2	3
3	11
4	-3
5	1
6	2
7	2
8	-3
9	1

J. Hoshen ar 38 (1976)

Algorithms

Hoshen and Kopelman



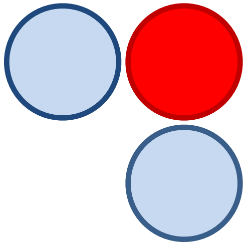
			3			9
7	7		8	3		
				3		6
		3	3	3		6
		3				
2		4	3		5	
2	2		3			

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

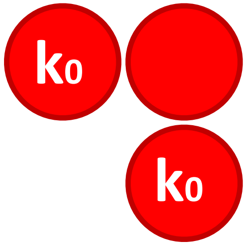
Hoshen and Kopelman



Isolated

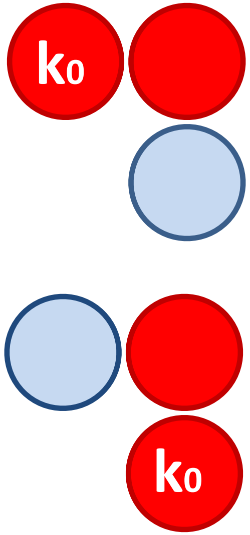
$k = k + 1$

$M(k) = 1$



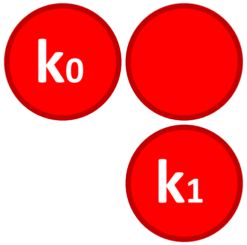
Two neighbor k_0:

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



One neighbor k_0:

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



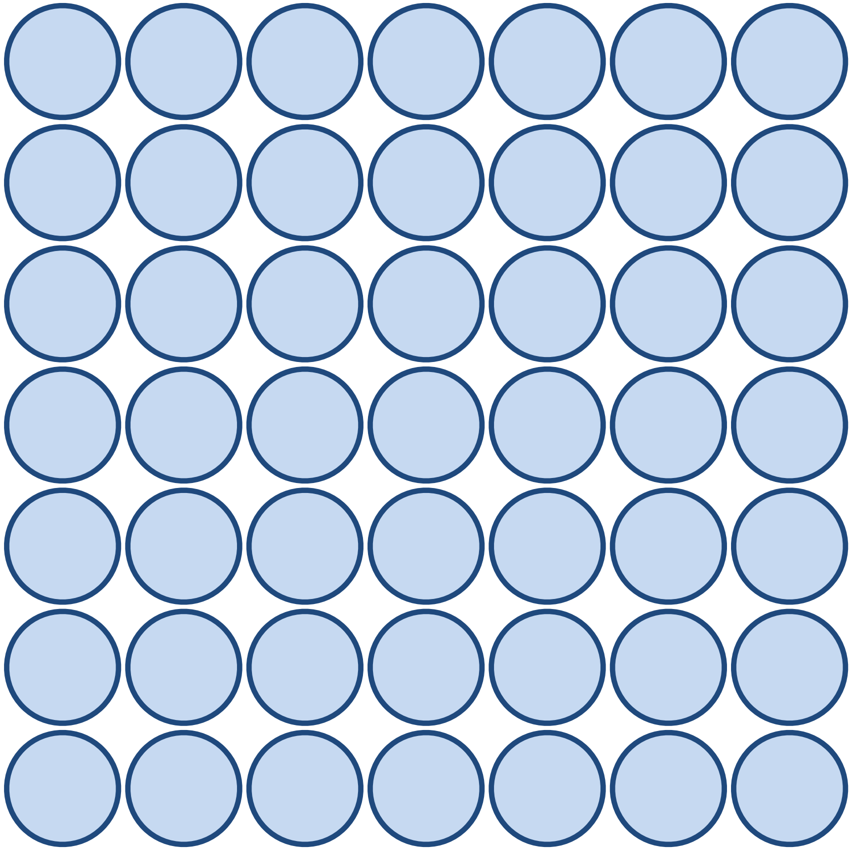
One neighbor k_0 and one neighbor k_1 :

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

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

Algorithms

Newman and Ziff (microcanonical)

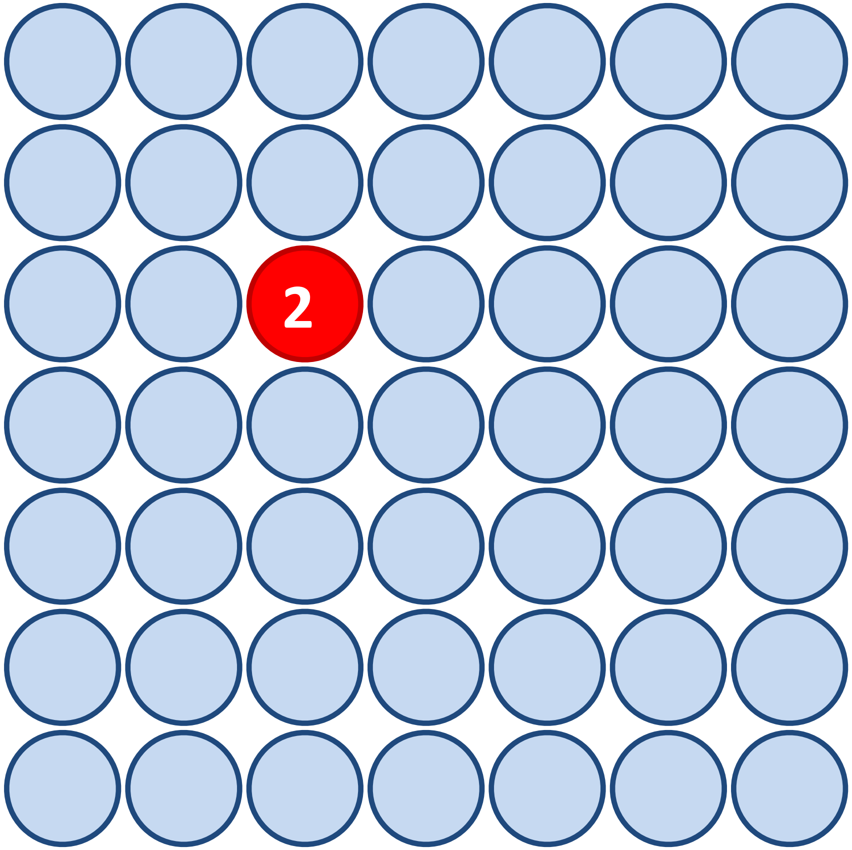


k	M(k)
2	0

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)

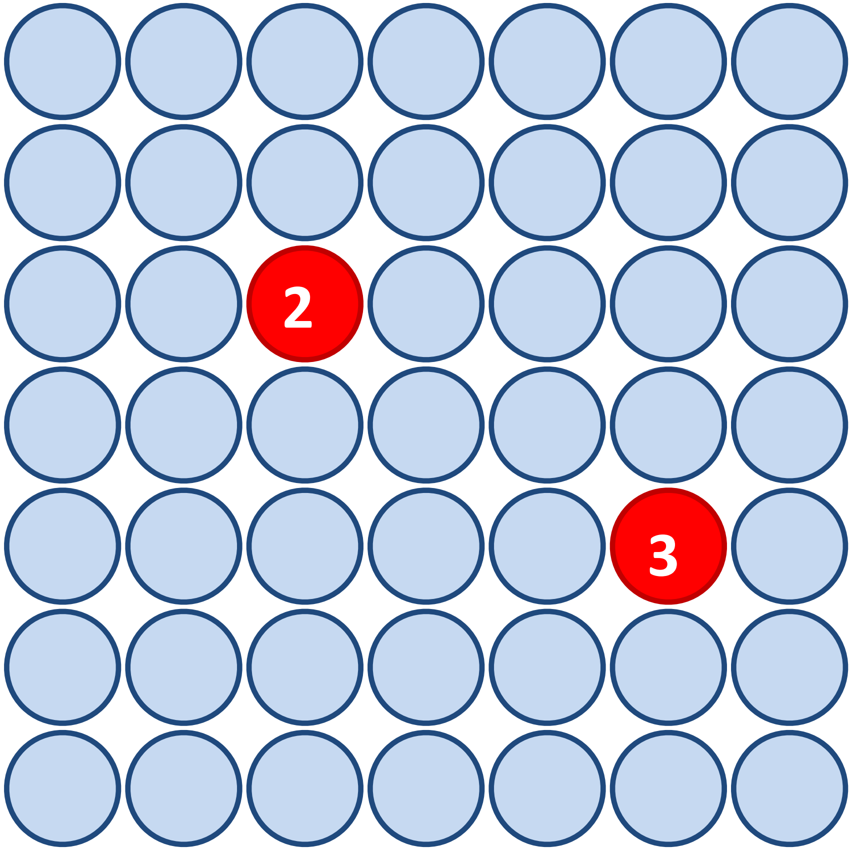


k	M(k)
2	1

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)

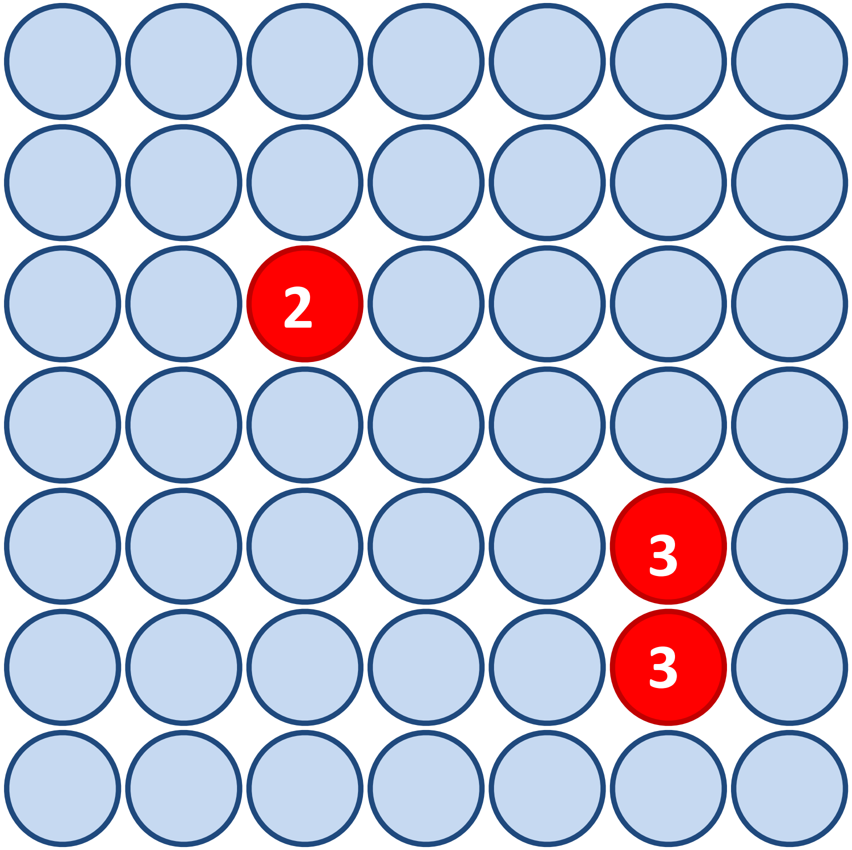


k	M(k)
2	1
3	1

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)

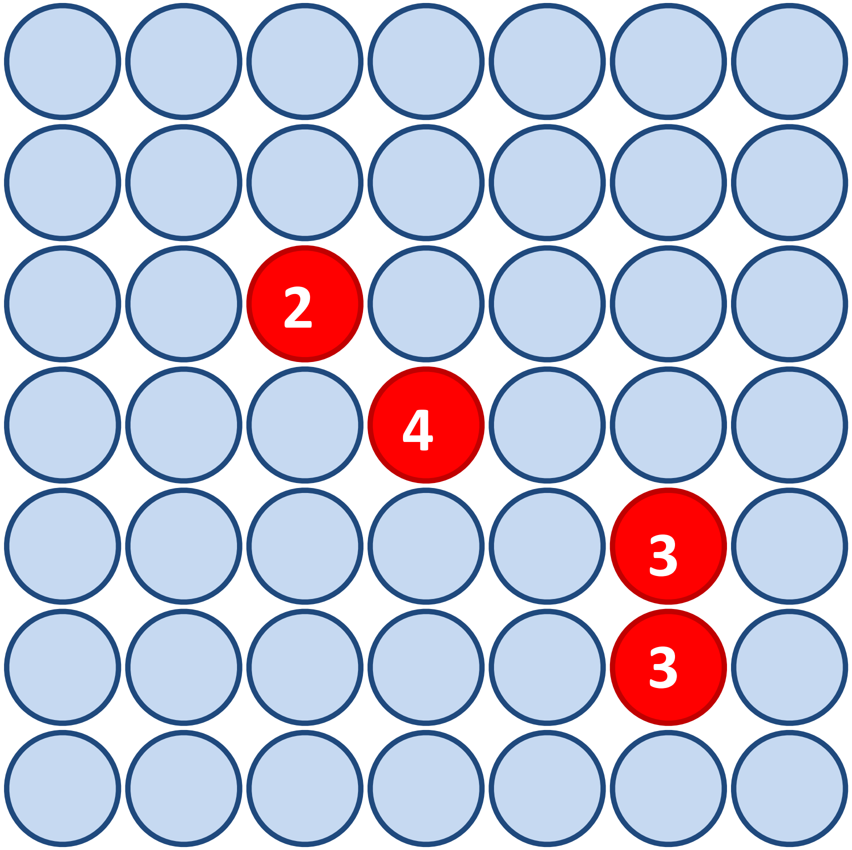


k	M(k)
2	1
3	2

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)

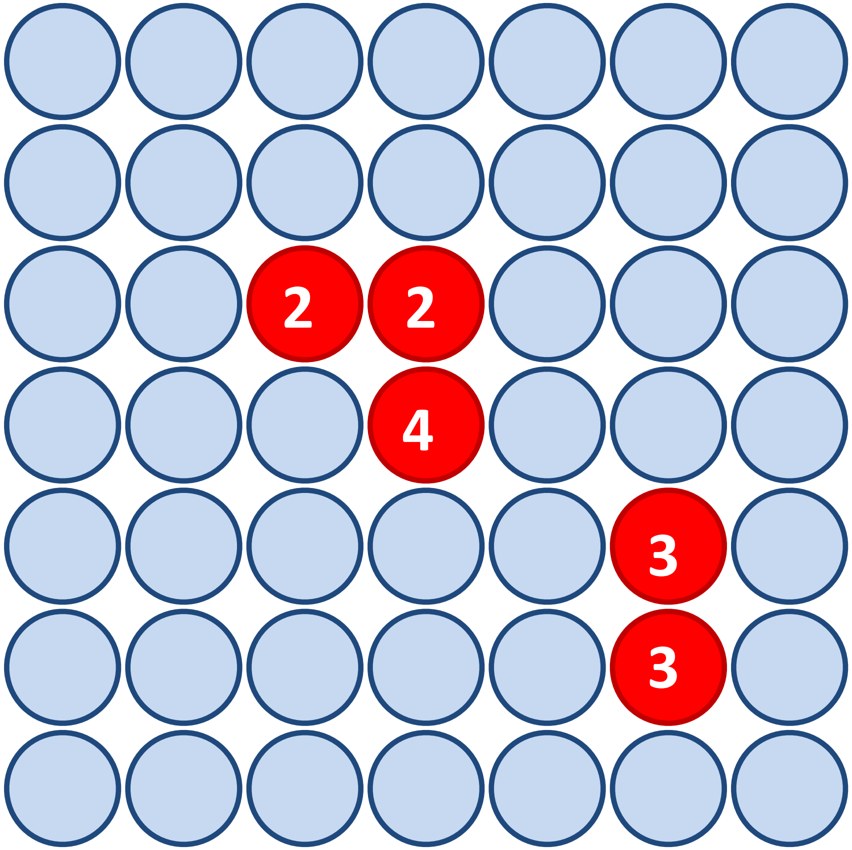


k	M(k)
2	1
3	2
4	1

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

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

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)