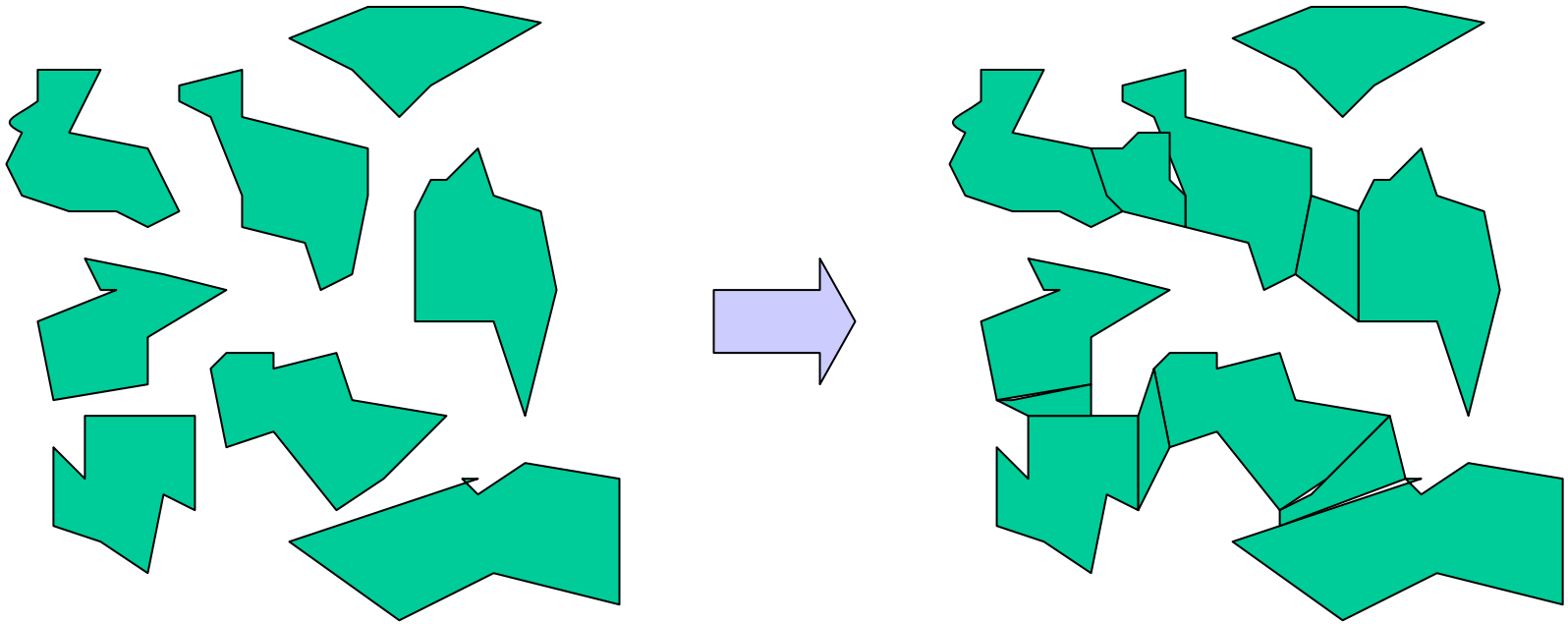
2．14　パーコレーション・クラスタリング

ゾル-ゲル転移：　ゼリー，豆腐製造，凝血

パーコレーション現象　＝　液体から準固体への遷移
→　分子がランダムにつながって一緒になる

パーコレーション転移：　多くの現象はランダムな島々からできていて，ある条件でこれらの島の間に1つの
巨視的な大陸が現れるようなもの



化学系: 重合反応
生物系: 免疫の抗原抗体反応
物理学: 臨界現象

○ ランダム・サイト・パーコレーション

相互作用の無い系のサイト・パーコレーション

m × m のランダムな論理格子: 0(空)か1(占有)の値

p: サイトが占有されている確率

1つのクラスター: 占有された最近接サイトの集まり

① m × m 正方格子の各サイトにランダムな値(a = Random[ ])を割当て

② a≦p → 1, a > p → 0

```
In[1] := SitePercolation[p_, m_Integer] :=
                Table[Floor[1+ p – Random[ ]], {m}, {m}]
```

```
In[2] := r = SitePercolation[0.35, 9]
Out[2] := {{0, 0, 1, 1, 0, 0, 0, 1, 0}, {0, 0, 1, 0, 0, 1, 0, 0, 0},
           {0, 0, 1, 0, 1, 1, 1, 0, 0}, {0, 0, 1, 1, 1, 1, 0, 0, 1},
           {1, 0, 0, 1, 1, 0, 0, 0, 0}, {0, 0, 1, 1, 1, 0, 0, 0, 0},
           {1, 0, 1, 1, 0, 1, 0, 1, 0}, {0, 1, 0, 0, 0, 1, 1, 1, 0},
           {0, 0, 0, 0, 0, 0, 0, 0, 0}}

In[3] := r //MatrixForm
Out[3]//MatrixForm = 0 0 1 1 0 0 0 1 0
                     0 0 1 0 0 1 0 0 0
                     0 0 1 0 1 1 1 0 0
                     0 0 1 1 1 1 0 0 1
                     1 0 0 1 1 0 0 0 0
                     0 0 1 1 1 0 0 0 0
                     1 0 0 0 0 1 0 1 0
                     0 1 0 0 0 1 1 1 0
                     0 0 0 0 0 0 0 0 0
```

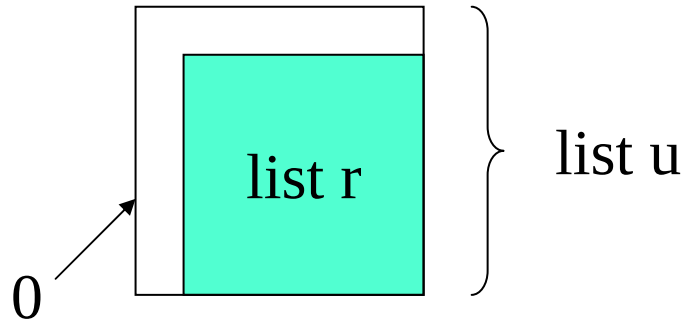
浸透閾値，臨界浸透確率（critical percolation probability）:
　１つの両端まで延びた（spanning）クラスター（格子を横切った途切れない道筋）が最初に起こる確率 p

○　クラスターのラベルづけ
　ホーシェン-コペルマン・アルゴリズム
　ネストしたリスト r のラベルづけ

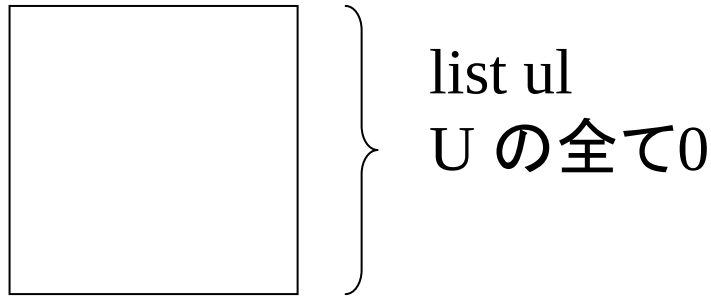
```
In[1] := ( r = {{1, 0, 0, 1}, {1, 0, 1, 1}, {1, 1, 1, 0}, {0, 0, 0, 1}}
                    //MatrixForm
Out[1]//MatrixForm = 1 0 0 1
                     1 0 1 1
                     1 1 1 0
                     0 0 0 1
```

list r

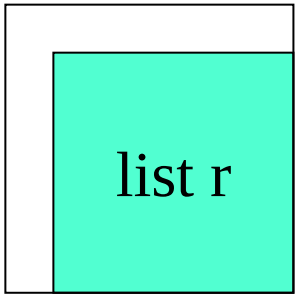
```
In[2] := ( u = {{0, 0, 0, 0, 0}, {0, 1, 0, 0, 1}, {0, 1, 0, 1, 1},
                {0, 1, 1, 1, 0}, {0, 0, 0, 0, 1}}//MatrixForm
Out[2]// MatrixForm =  0 0 0 0 0
                       0 1 0 0 1
                       0 1 0 1 1
                       0 1 1 1 0
                       0 0 0 0 1
```




```
In[3] := (ul= u /. -> 0 )//MatrixForm
Out[3]//MatrixForm = 0 0 0 0 0
                     0 0 0 0 0
                     0 0 0 0 0
                     0 0 0 0 0
                     0 0 0 0 0
```




```
In[4] := ulp = { }
```

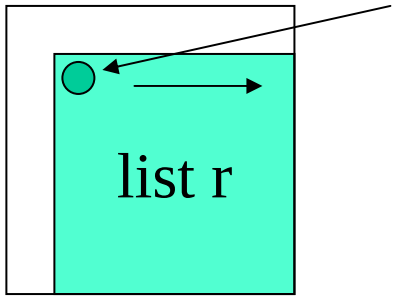


list *u*

クラスターにラベルをつけるルール

list *ul*

○　ルールの概要

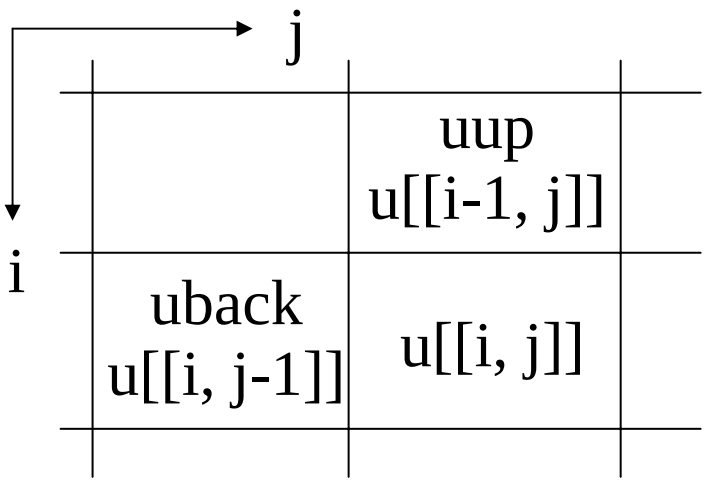


①　ここから開始する

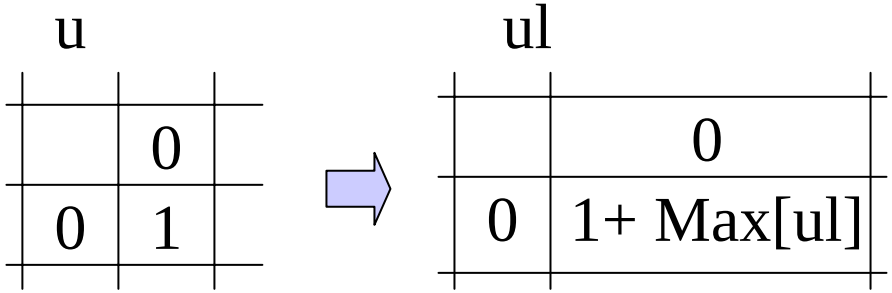
②　占有されたサイトについて
　左および上のサイトとの繋がりを見る.
2-1　繋がりが無ければ，新しいラベルを付ける.
2-2　繋がりがあれば，そのラベルをつける.

ルール

A: u[[i, j]] = 0 → 次のサイト



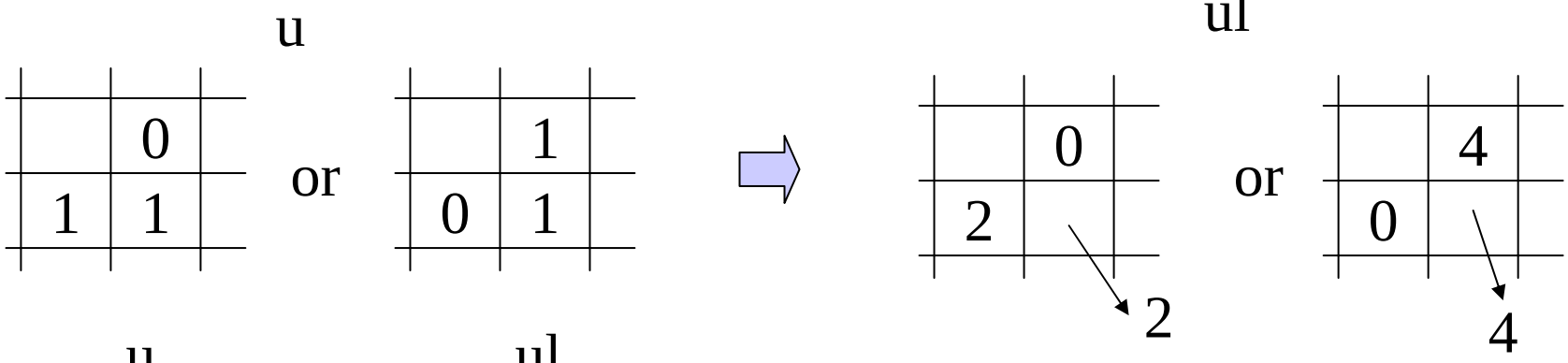
B:



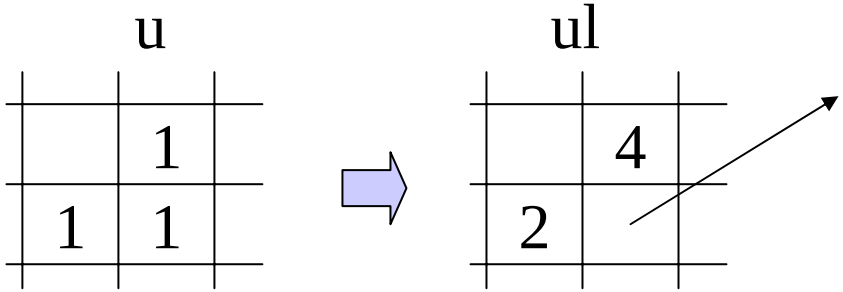
繋がっていない

ulp に新しい最大値を加える

C:

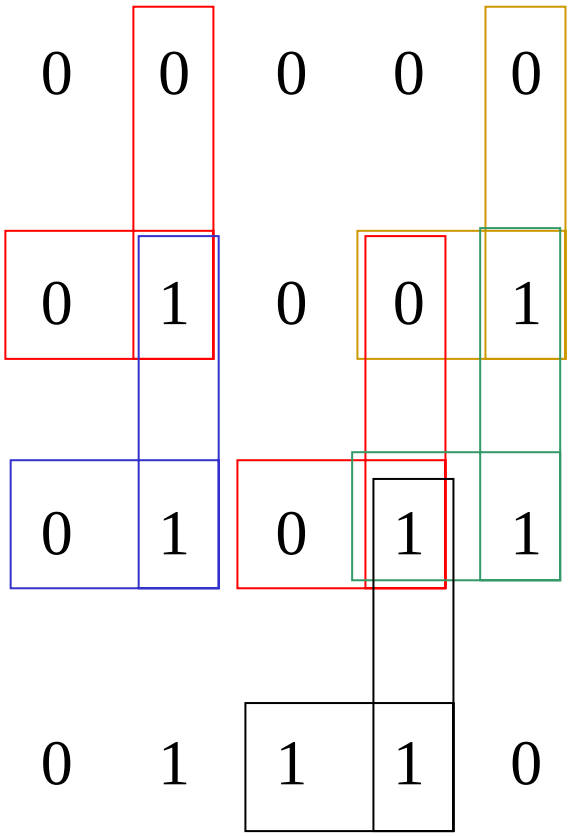


D:



小さいラベル 2

ulp[[ulup]] の値4 → 2



ul

0 0 0 0 0

0 0→1 0 0 0→1+Max[ul] = 2

ulp={1} {1, 2}

0 0→1 0 0→3 0→2

{1, 2, 3}

→2→1

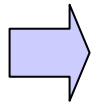
0 0 0 0 0

0 0 0 0 0

0 0 0 0 0
0 1 0 0 1
0 1 0 1 1
0 1 1 1 0
0 0 0 0 1

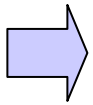
マトリクス *u*

0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

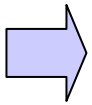


0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

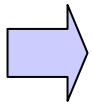
マトリクス *ul*のスタート



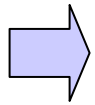
0 0 0 0 0
0 1 0 0 2
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0



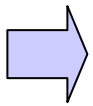
0 0 0 0 0
0 1 0 0 2
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0



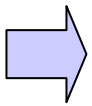
0 0 0 0 0
0 1 0 0 2
0 1 0 3 0
0 0 0 0 0
0 0 0 0 0



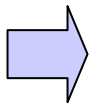
0 0 0 0 0
0 1 0 0 2
0 1 0 2 2
0 0 0 0 0
0 0 0 0 0



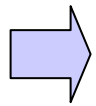
0 0 0 0 0
0 1 0 0 2
0 1 0 2 2
0 1 0 0 0
0 0 0 0 0



0 0 0 0 0
0 1 0 0 2
0 1 0 2 2
0 1 1 0 0
0 0 0 0 0

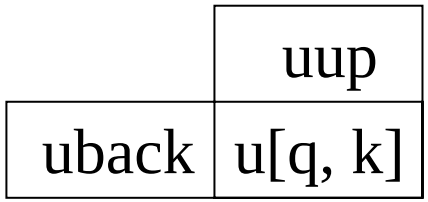


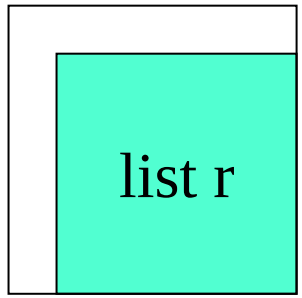
0 0 0 0 0
0 1 0 0 1
0 1 0 1 1
0 1 1 1 0
0 0 0 0 0



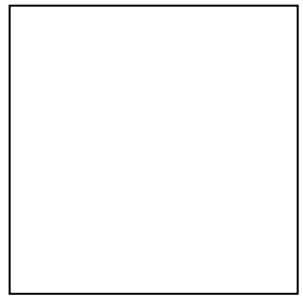
0 0 0 0 0
0 1 0 0 1
0 1 0 1 1
0 1 1 1 0
0 0 0 0 4

プログラム



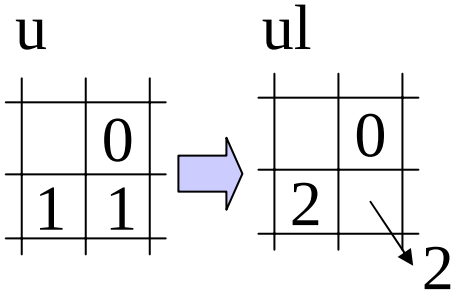


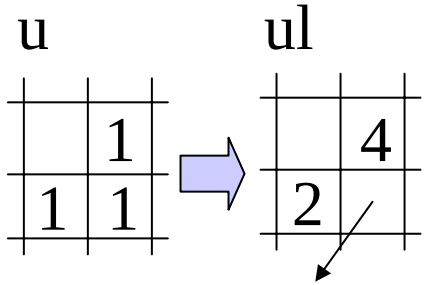
list *u*



list *ul*

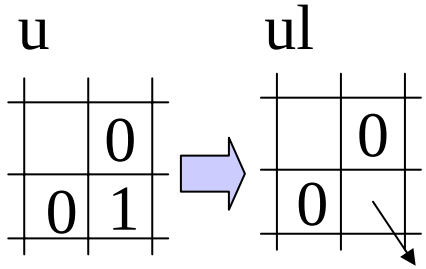
```
In[1] := ClusterLabel[r_List] :=
          Module[{uup := u[[q-1, k]], uback := u[[q, k-1]],
                  ulup := ul[[q-1, k]], ulback := ul[[q, k-1]]},
                 AddZeros[w_] :=
                  Prepend[w, Table[0, {Length[w[[1]]}]];
                 u = Transpose[AddZeros[Transpose[AddZeros[r]]]];
                 ul = u/. 1 -> 0;
                 ulp = { };
```





小さいラベル 2

ulp[[ulup]] の値4 → 2



1+ Max[ul]

```
Do[
   which[u[[q, k]] = = 1 && uup = = 1 &&
            uback = = 1 && ulup =!= ulback,
        ul[[q, k]] = Min[ulp[[ulback]], ulp[[ulup]]];
        ulp[[Max[ulp[[ulback]], ulp[[ulup]]]]] =
             Min[ulp[[ulback]], ulp[[ulup]]],
     u[[q, k]] = = 1 && uup = = 1, ul[[q, k]] = ulup,
     u[[q, k]] = = 1 && uback = = 1, ul[[q, k]] = ulback,
     u[[q, k]] = = 1, ul[[q, k]] = Max[ul] + 1;
        AppendTo[ulp, Max[ul]]
   ],
```

```
  {q, 2, Length[u]}, {k, 2, Length[u]}
  ];

new = Range[Length[ulp]];
relabelrules1 = Thread[new -> ulp];
correctlabels = ulp//. Relabelrules1;
relabelrules2 = Thread[Union[correctlabels] ->
          Range[Length[Union[correctlabels]]]];
ufinal = ul //. relabelrules1 /. relabelrules2
]
```

ClusterLabel + SitePercolation

→ 簡単なランダム・サイト・パーコレーション系を作り、ラベルづけ

（ｉ）表で表示

```
In[1]:= r =SitePercolation[0.493, 9];
        (cl = ClusterLabel[r] //TableForm
```

（ii）明暗で表示

```
In[2]:= ListDensityPlot[Reverse[cl],
                        ColorFunction -> GrayLevel,
                        Frame -> False]
```

（iii）彩色で表示

```
In[3]:= ShowPercolation[cluster_List, opts___]:= Module[{},
            Show[Graphics[RasterArray[
                    Reverse[Map[Hue[# Random[]/Max[cluster]]&,
                                cluster, {2}]]] ],
                 opts, AspectRatio -> 1]]
```

［課題］

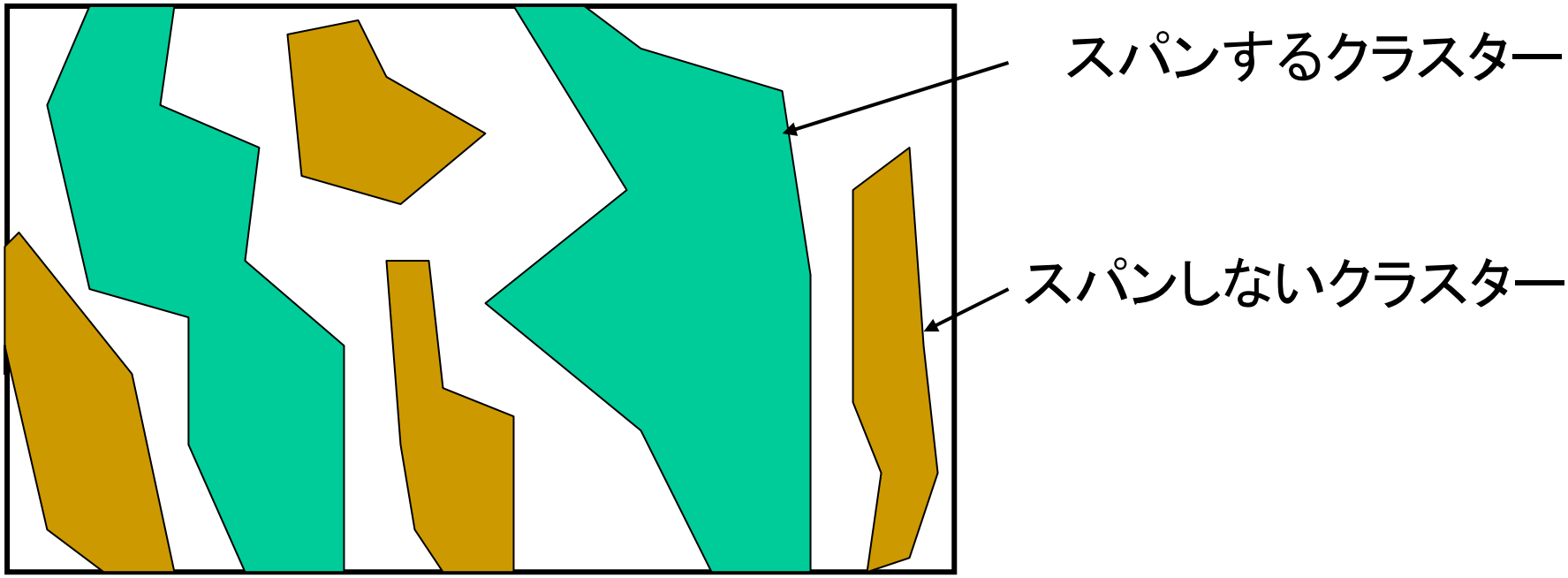
（ｉ）浸透閾値 p_c ：1つのクラスターが格子をスパンする（格子の両端に延びる） p

（ii）p_s ：1つの占有サイトが1つのスパンするクラスターに属する p

$p_s = n_{span}/n_{occupied}$

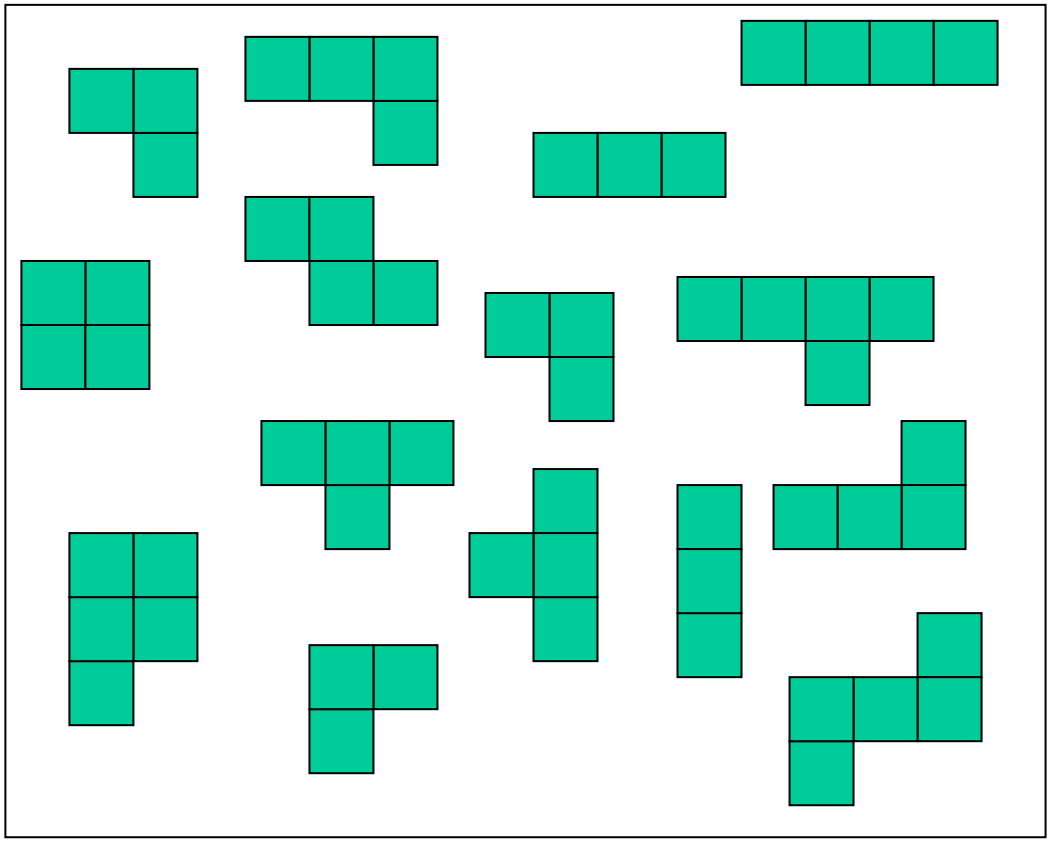
n_{span} ：スパンするクラスターにあるサイト数

$n_{occupied}$ ：占有サイトの全数



（iii）平均クラスターサイズ＝ $\sum_s s^2 n_s \Big/ \sum_s s\, n_s$

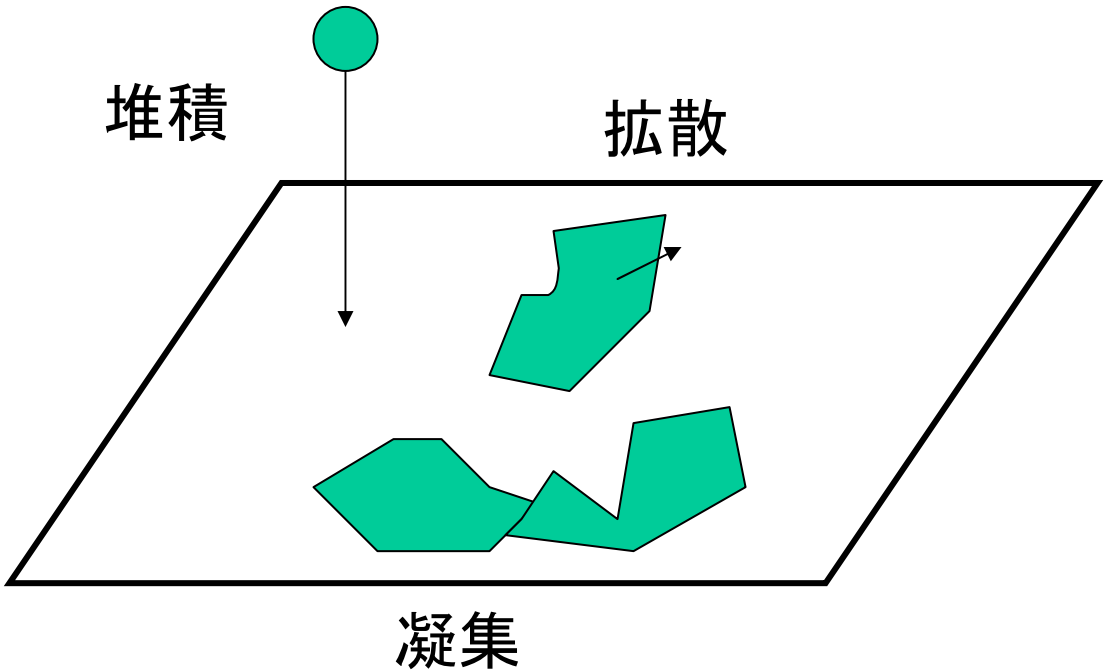
$n_s : s$ サイトをもつクラスター

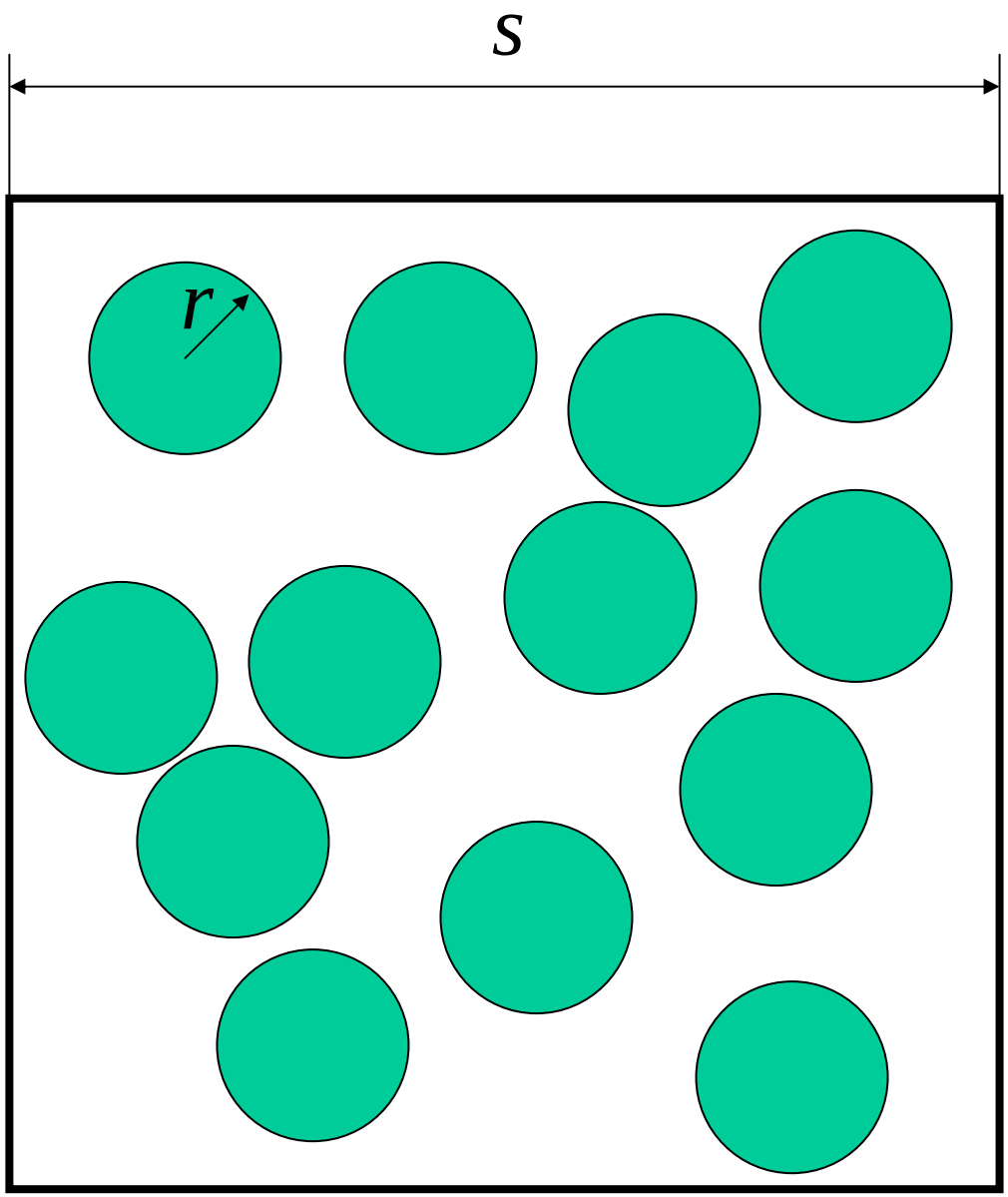


s	n_s
3	5
4	7
5	3

[ナノ構造形成のモデル]

(1) 堆積：粒子が正方格子の上にランダムに置かれる
(2) 拡散：各時間ステップで1つのクラスター（粒子が繋がったもの）
　　確率pで上下左右に1単位動く
(3) 凝集：粒子が隣接すれば、クラスターは一緒にくっつく





n 個

端から端まで途切れなく繋がる個数

```
SwissCheese[s_Integer, n_Integer, r_]:=
  Module[{adddisks = 1},
      diskcount = n;
      t = Random[Real, {.15, .5}];
      box = Graphics[{RGBColor[.76, .61, .07],
                                  Rectangle[{0, 0}, {s, s}]}];
      disklocs = Table[{Random[Real, s, 4],
                               Random[Real, s, 4], {n}];
      disks := ListPlot[disklocs,
                  PlotStyle -> {RGBColor[0.65, 0.08, 0.7],
                  PointSize[t/s]},
                  Axes -> None,
                  DisplayFunction -> Identity];
```

```
    While[adddisks != 0,
            Show[box, disks,
                  DisplayFunction->$ DisplayFunction];
            Print[“total number of disks =  ” , diskcount];
            Print[“disk density = ”,
                  (N[Pi])(r^2)(diskcount)/(s^2)];
            Print[“PointSize = ”, t];
                  adddisks = Input[“How many more disks?”];
            diskcount += adddisks;
            disklocs = Join[disklocs,
                              Table[{Random[Real, s, 4],
                                    Random[Real, s, 4]},
                              {adddisks}]]
    ]
]
```