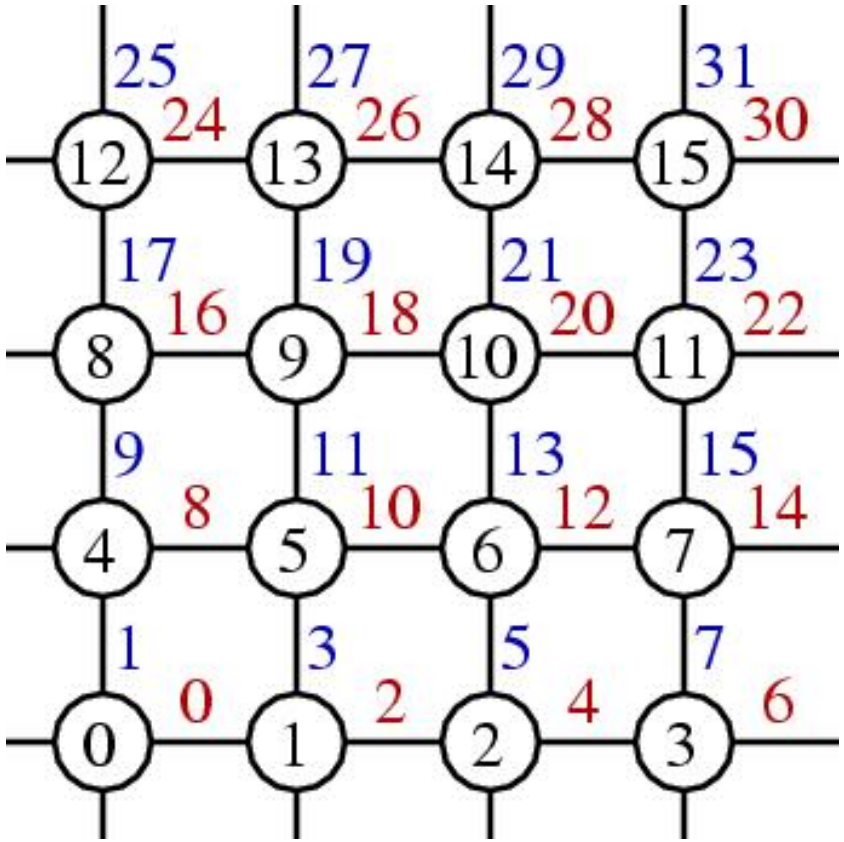
Programming the Swendsen-Wang algorithm

To construct clusters, we need arrays containing

- Neighbors of given site s: `neighbor[i,s]`
- Two spins connected by given bond b: `bondspin[i,b]`
- Bonds connected to given spin s: `spinbond[i,s]`

Labeling of spins and bonds; example in 2D



Note: in Julia the labels have to start from 1, adjust accordingly

Storing spin and bond variables in one-dimensional vectors

```
spin[1:n], bond[1:d*n]
```

Construction of lattice arrays in 2D

```
subroutine lattice

for s0=1:n
   x0=mod(s0-1,ll)
   y0=div(s0-1,ll
   x1=mod(x0+1,ll)
   x2=mod(x0-1,ll)
   y1=mod(y0+1,ll)
   y2=mod(y0-1,ll)
   s1=1+x1+y0*ll
   s2=1+x0+y1*ll
   s3=1+x2+y0*ll
   s4=1+x0+y2*ll
   neighbor[1,s0]=s1
   neighbor[2,s0]=s2
   neighbor[3,s0]=s3
   neighbor[4,s0]=s4
   bondspin[1,2*s0]=s0
   bondspin[2,2*s0]=s1
   bondspin[1,2*s0+1]=s0
   bondspin[2,2*s0+1]=s2
   spinbond[1,s0]=2*s0
   spinbond[2,s0]=2*s0+1
   spinbond[3,s1]=2*s0
   spinbond[4,s2]=2*s0+1
end do
```

Main program

```
bprob=1.d0-exp(-2.d0/temp)
 for i=1:div(steps,4)
    castbonds()
    flipclusters()
 end
 for j=1:bins
    resetbindata()
    for i=1:steps
       castbonds()
       flipclusters()
       measure()
    end
    writebindata(n,steps)
 end
```

Generating bond configuration

```
function castbonds()

for b=1:2^n
   if spin[bondspin[1,b]]==spin[bondspin[2,b]]
      if ran()<=bprob
         bond[b]=true
      else
         bond[b]=false
      end
   else
      bond[b]=F
   end
end
```

For cluster finding/flipping, see program sw.jl