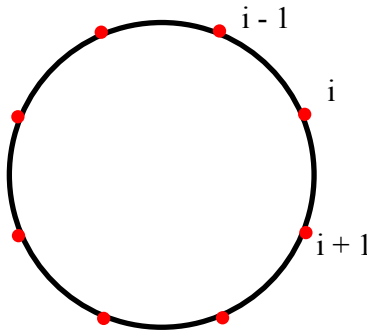
Spotting The Correct Algorithm

Consider a one dimensional system of N sites with periodic boundary conditions, with a single particle hopping between neighboring sites at time $t = 0, 1, 2, \ldots$ as shown in the next figure. A move from site i to site $i+1$ modulo N is "to the right" and a move from site i to site $i-1$ is "to the left".



The two following programs implement the **Markov-chain Monte Carlo** algorithm such at that each time step, the particle moves with probability 1/2 to the right and with probability 1/2 to the left. We also have to note that the algorithm (in both programs) satisfies detailed balance with a constant probability on all sites and that it is irreducible thus it cannot be broken up into two independent processes.

```
import random
N = 20
position = 0
for t in range(1e5):
        direction = random.choice([-1,1])
        position  = (position + direction) % N
```

```
import random
N = 20
position = 0
for t in range(1e5):
        if random.uniform(0.0,1.0) < 0.5:
                position = (position + 1) % N
        else:
                position = (position - 1) % N
```