Cyclic Barrier

Creating a barrier

Number of threads that need to reach it in order to allow access

```
CyclicBarrier barrier = new CyclicBarrier(5);
```

Block until the required number of threads call this method

```
try {
        barrier.await();
} catch (InterruptedException | BrokenBarrierException
e) {
        e.printStackTrace();
}
```

Cyclic Barrier

- A barrier guarantees that all code above it is executed by all threads before any operation in the code below it

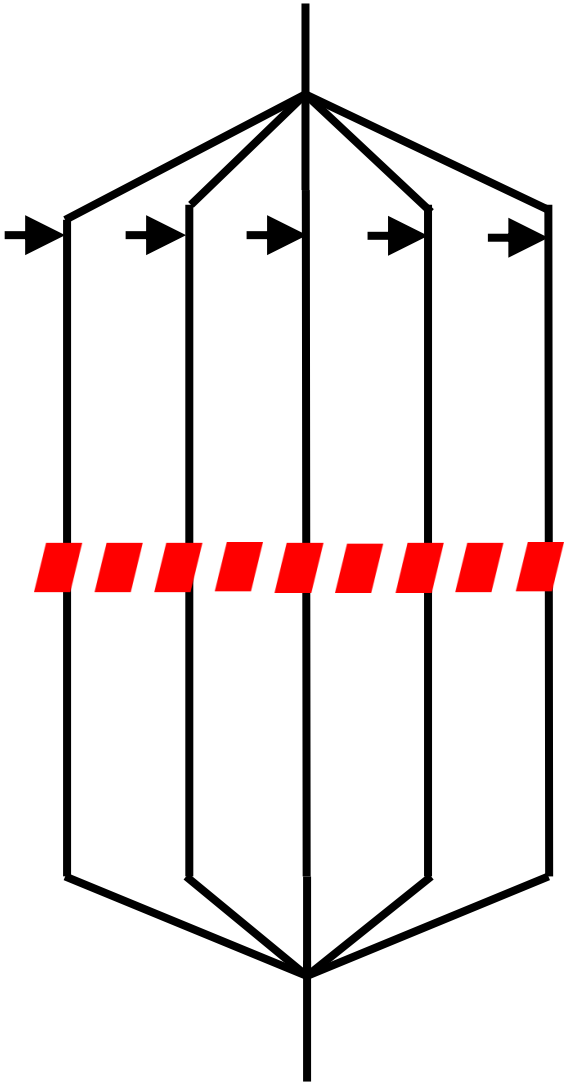
- Classic barrier can**not** be used inside a loop
  - Once all threads reach the barrier a thread that passes it can execute the entire code of the iteration and reach the barrier again before all threads pass, thus passing the barrier twice

- A CyclicBarrier can be safely used inside loops

Cyclic Barrier

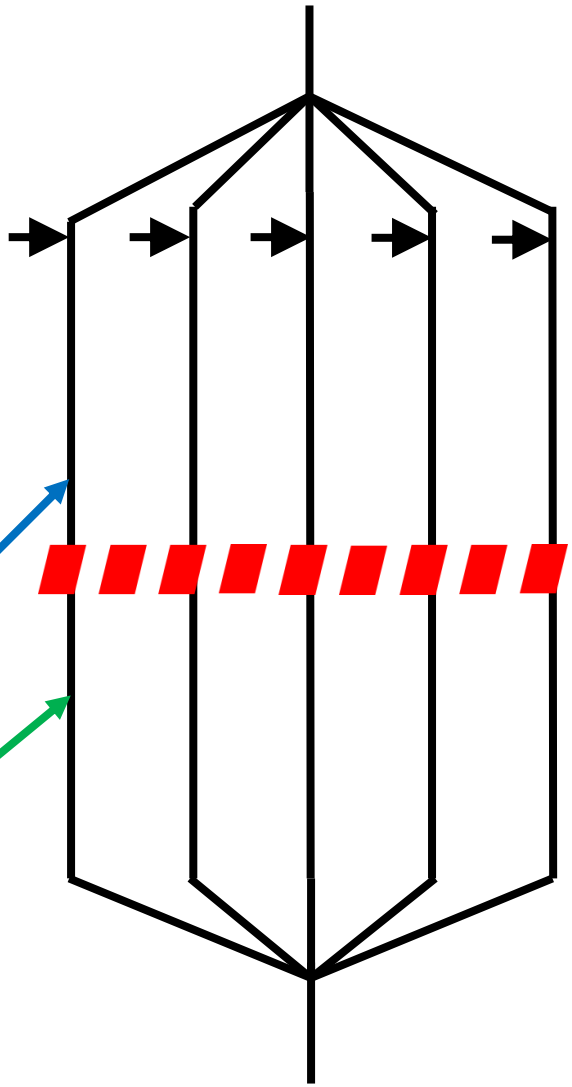
```
barrier.await();
```

5



Cyclic Barrier

```
barrier.await();
```
5

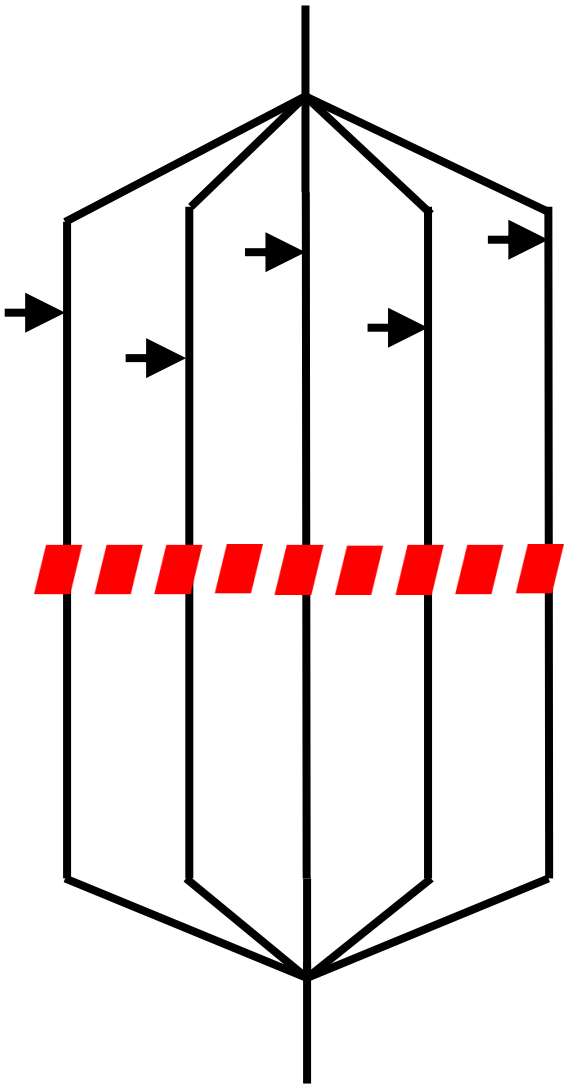


For all threads, all code here
is executed before any code here

Cyclic Barrier

```
barrier.await();
```

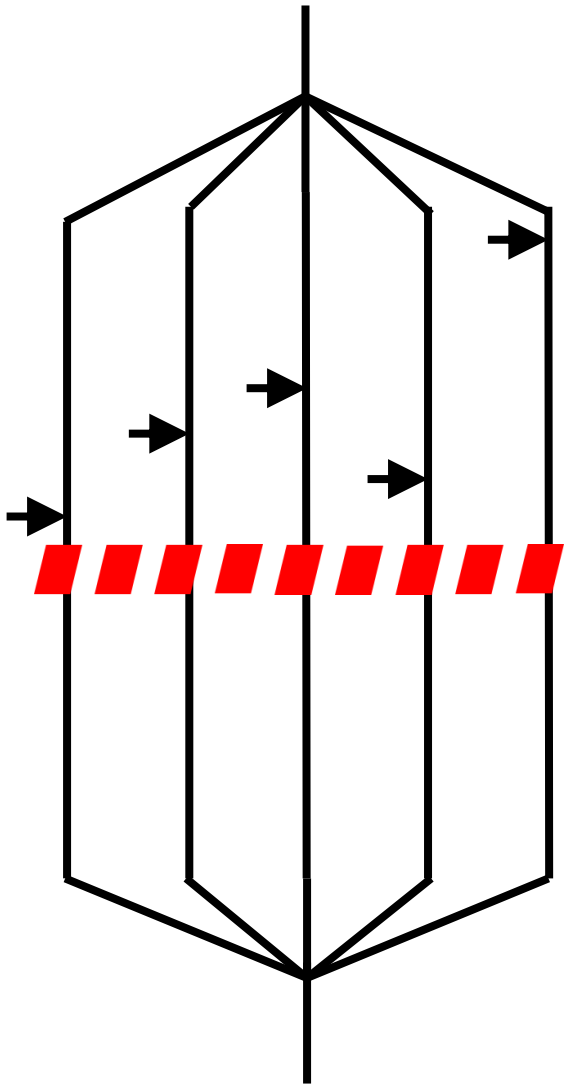
5



Cyclic Barrier

```
barrier.await();
```

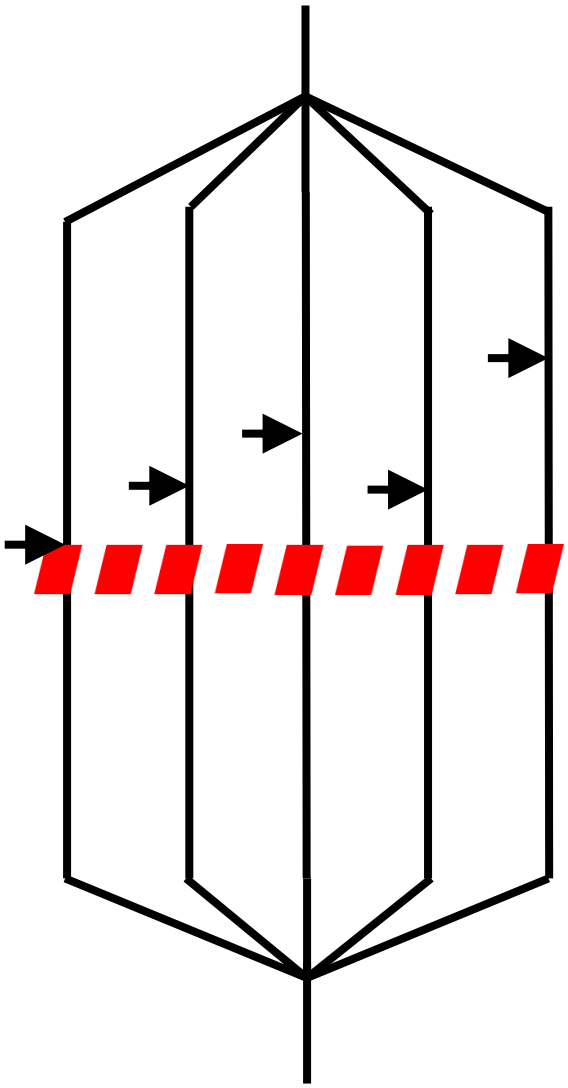
5



Cyclic Barrier

```
barrier.await();
```

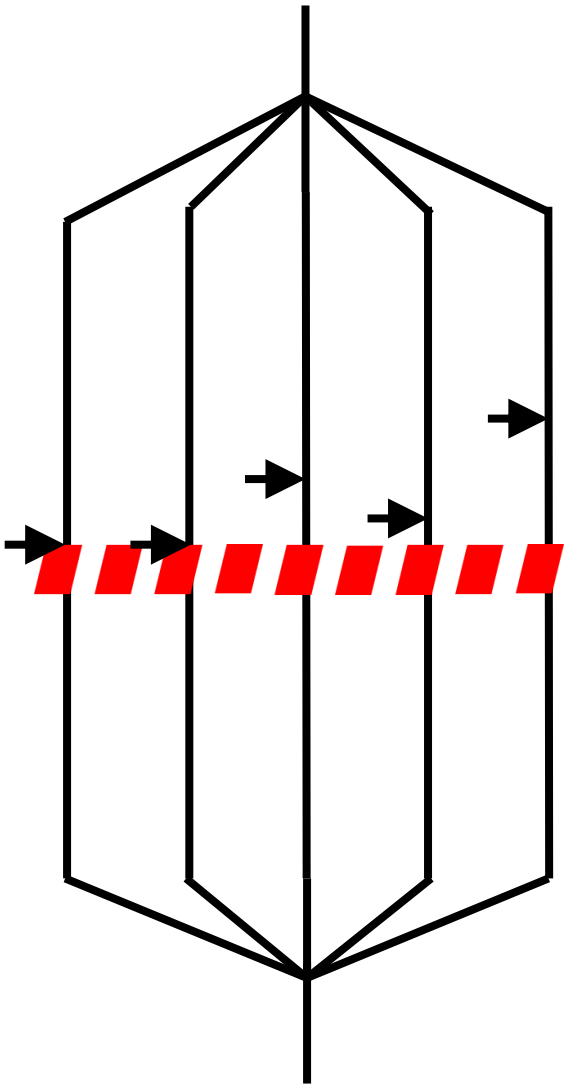
4



Cyclic Barrier

```
barrier.await();
```

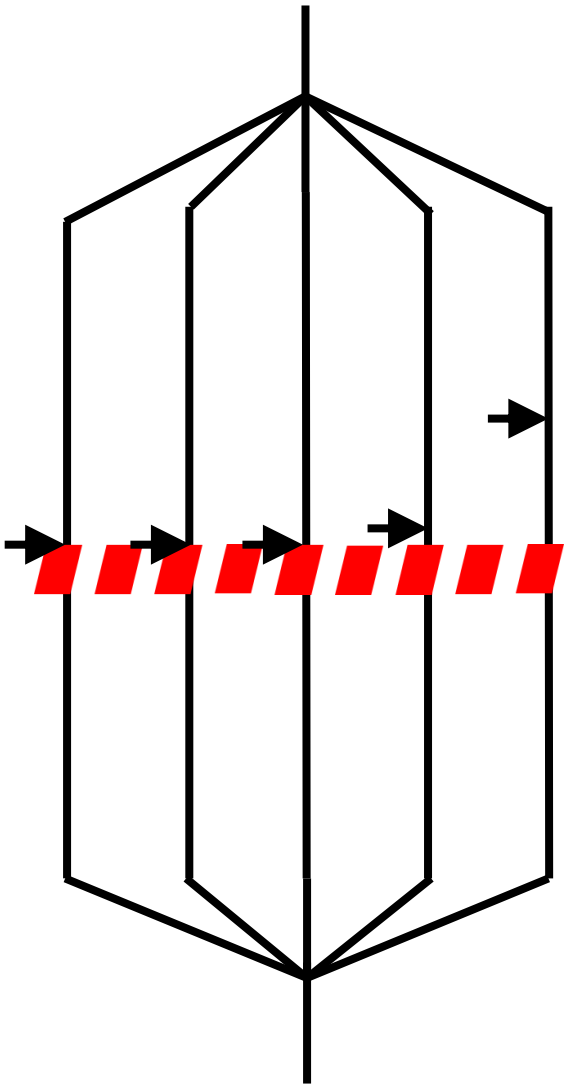
3



Cyclic Barrier

```
barrier.await();
```

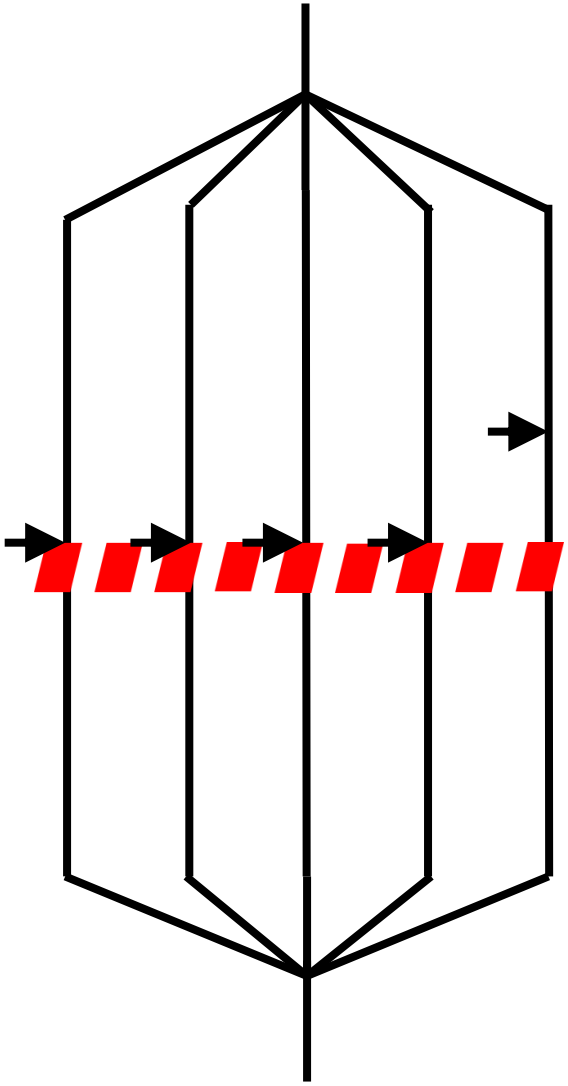
2



Cyclic Barrier

```
barrier.await();
```

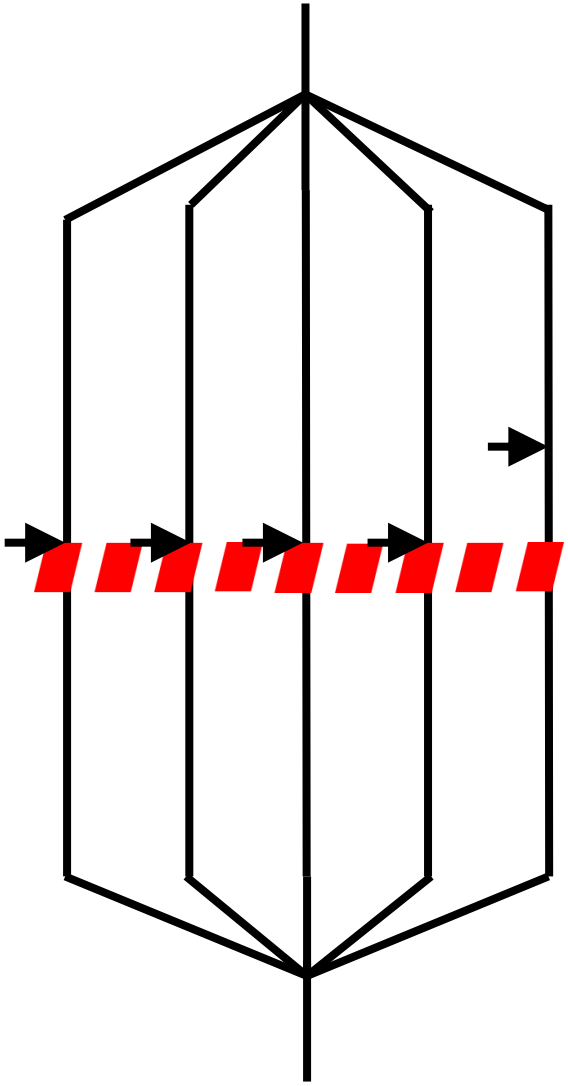
1



Cyclic Barrier

```
barrier.await();
```

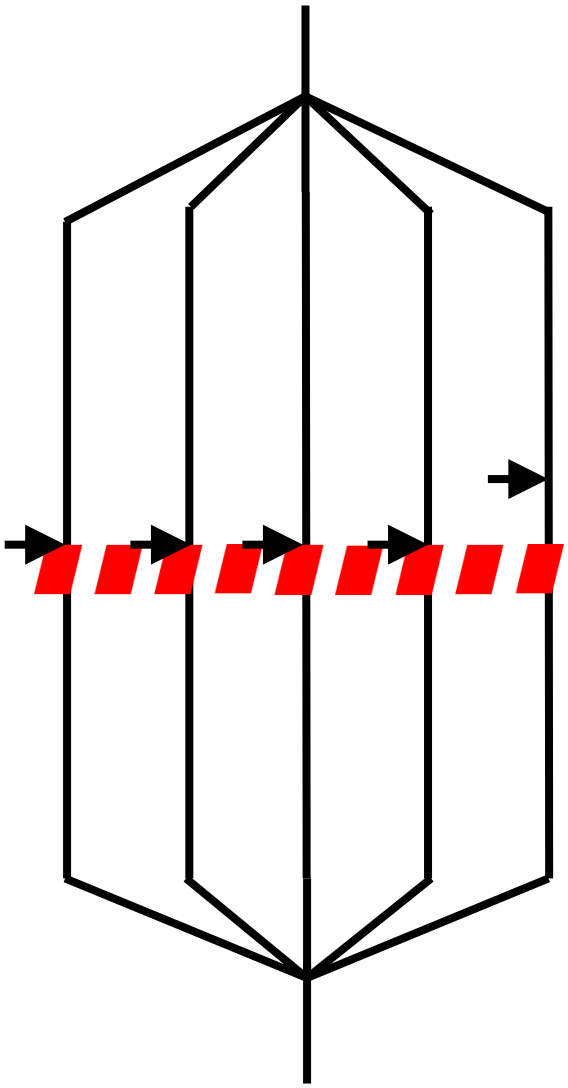
1



Cyclic Barrier

```
barrier.await();
```

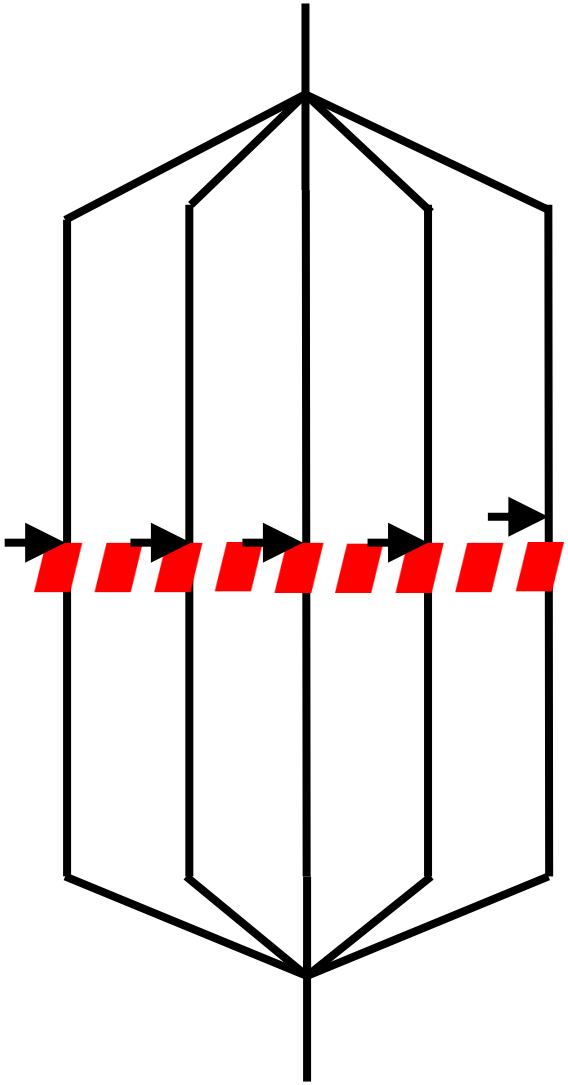
1



Cyclic Barrier

```
barrier.await();
```

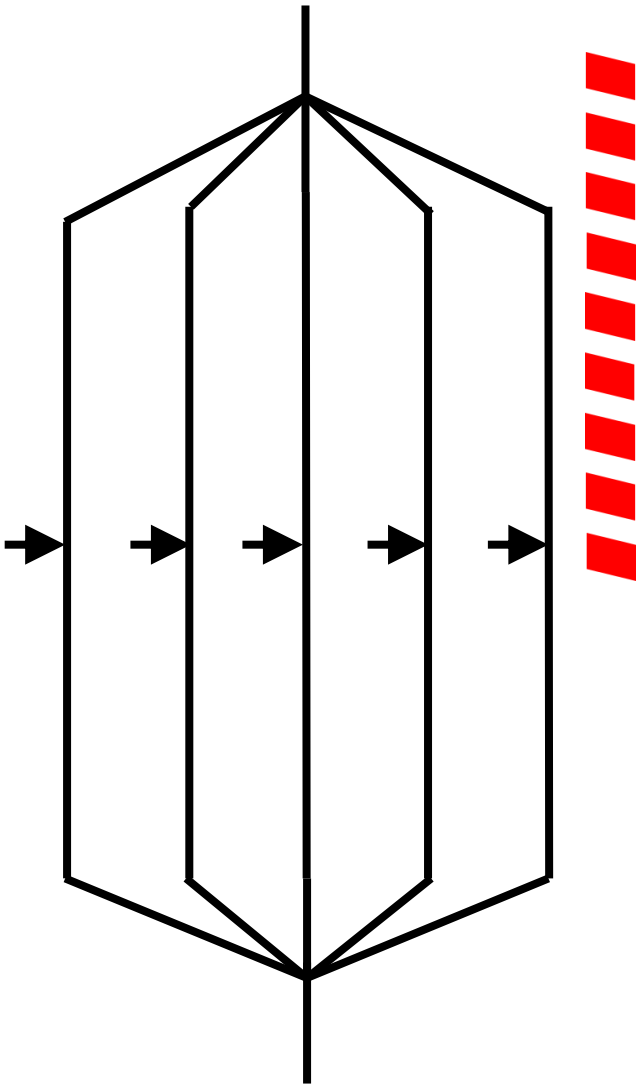
1



Cyclic Barrier

```
barrier.await();
```

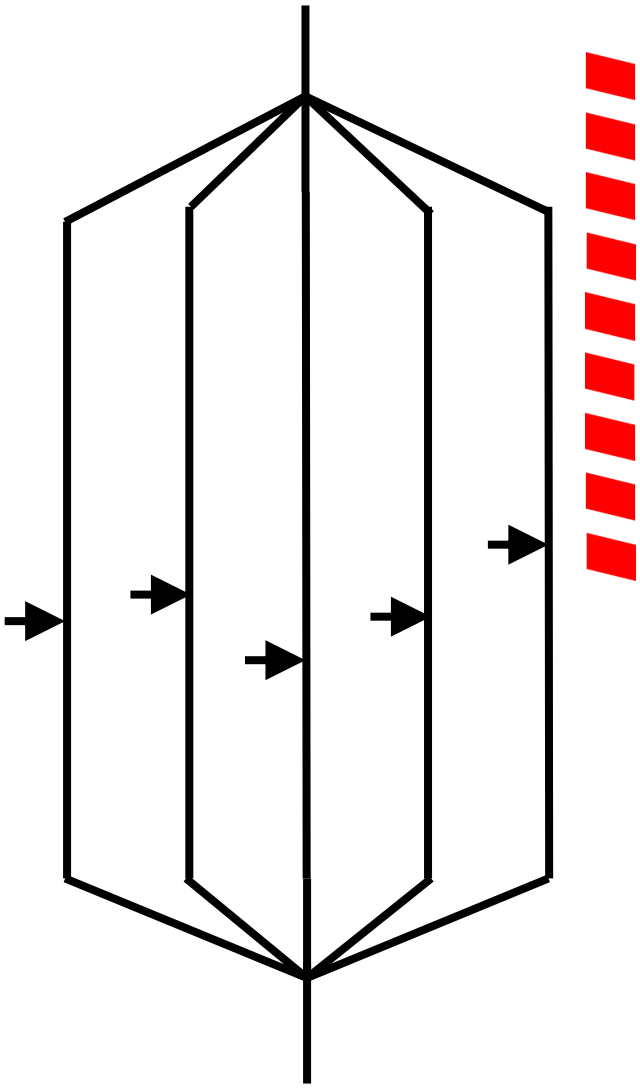
0



Cyclic Barrier

```
barrier.await();
```

0



Cyclic Barrier

How does the barrier know when to reset?

One possible solution is:
Reusable Barrier in
The Little Book of Semaphores
By Allen B. Downey

```
barrier.await();
```

5