libuv → event loop

Promises

Nodejs → official docs

① Ticks → one full trip of event loop.

Single iteration

JOIN THE DARKSIDE

Promises ticks timers I/O callback poll check close

ticks → process.nextTick → this func" takes a callback.

By calling process.nextTick we instruct node to invoke this cb func" at the end of curret operation before the next tick starts.

JOIN THE DARKSIDE

globals

process.env

env variable

process

diff properties assoc to your own process.

require

module

module pattern

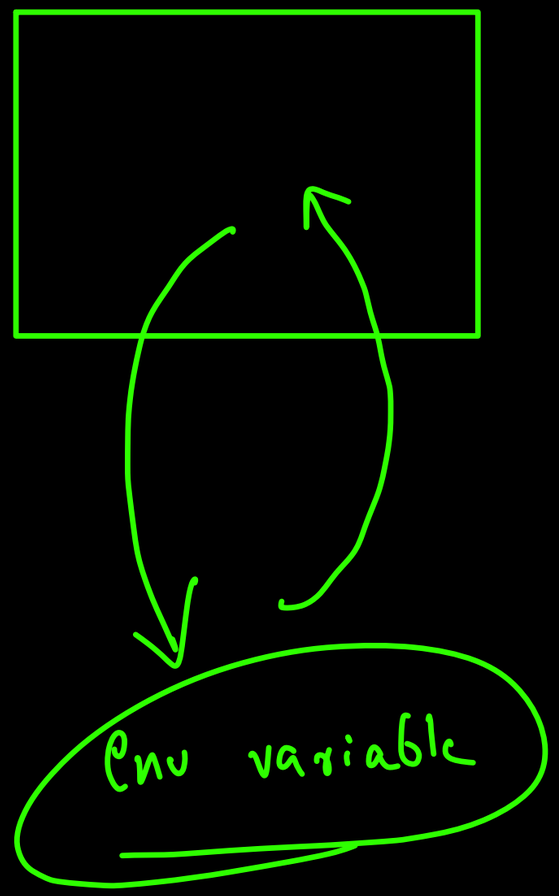
fs

file system

code → process

env variable

JOIN THE DARKSIDE


Env variable

↳ 1) → execution of nextTick callbacks.

↳ all of the cb are added one by one in the nextTick queue.

↳ whatever cb are there in the nextTick queue are one by one executed.

Priority nextTick queue > Priority Microtask queue.

JOIN THE DARKSIDE

a = 10

Promise.resolve( ).thu(() => ____);

cb

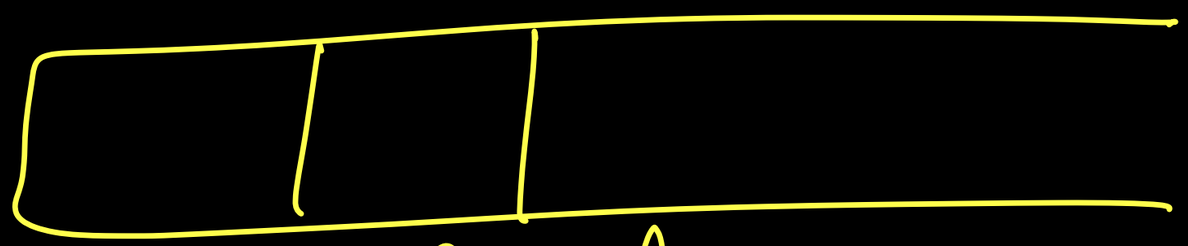
process.nextTick(() => ____);

cb1

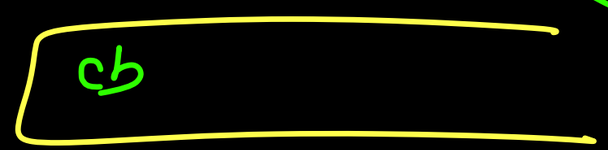
console.log(a)

cb1 cb2 cb3

10



↳ next tick Queue



microtask que

S: fulfilled
value: undefined
onfulfill: [ ]

event loop

JOIN THE DARKSIDE

→ The first iteration of event loop starts when your current set of instructions are done (main thread is done), and nextTick queue is empty, then the iteration of event loop start.

JOIN THE DARKSIDE

1. nextTick Queue will be resolved
2. Event loop tick (iteration starts)
3. Microtask queue / Promise queue (the queue who promise cb are present) starts exec.

4. Timers queue (contain all the callbacks of timers)
Starts exec

Start
Next Pick 1
Promise 1

JOIN THE DARKSIDE

```
console.log("Start");

setTimeout(() ⇒ {
    process.nextTick(() ⇒ console.log("Next Tick 2")); // cb will be waiting in nextTick queue
}, 0); // cb will be waiting in timer queue

Promise.resolve().then(() ⇒ console.log("Promise 1")); // cb will be waiting in promise queue

process.nextTick(() ⇒ console.log("Next Tick 1")); // cb will be waiting in nextTick queue

setTimeout(() ⇒ console.log("Timer 2"), 0); // cb will be waiting in timer queue
```

cb1 cb5 cb2 cb3 cb4

Start → N ~ T ~ 1

Nodejs → libuv

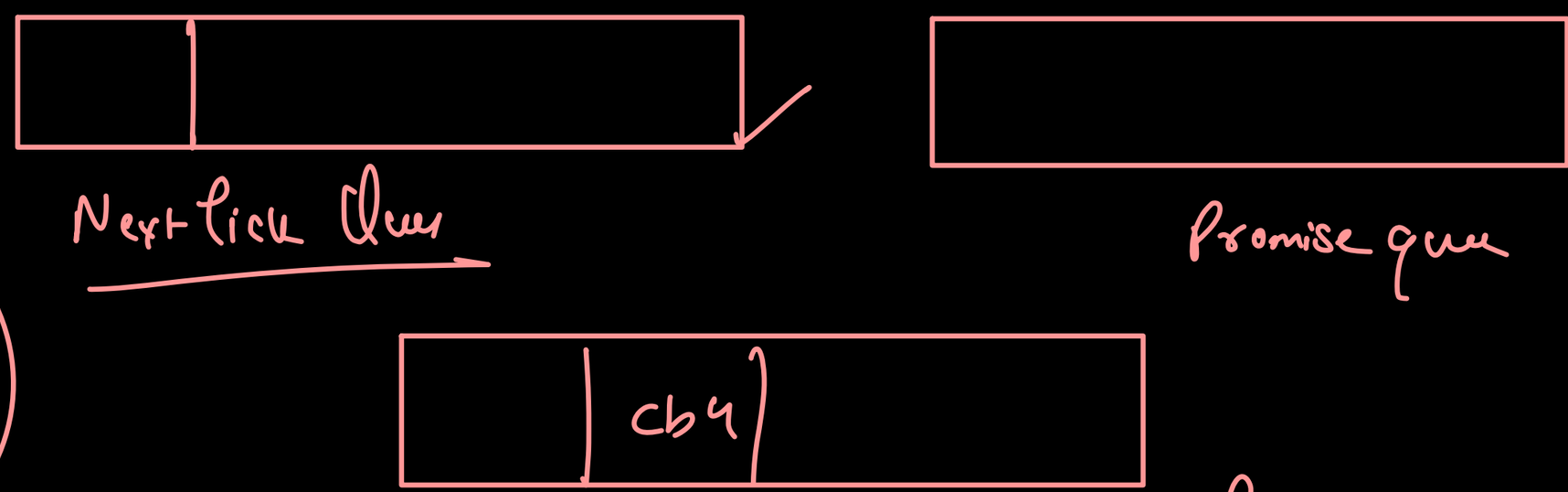
time → 1ms

time → 1ms

Cb3 Cb5

cb2

Cb1

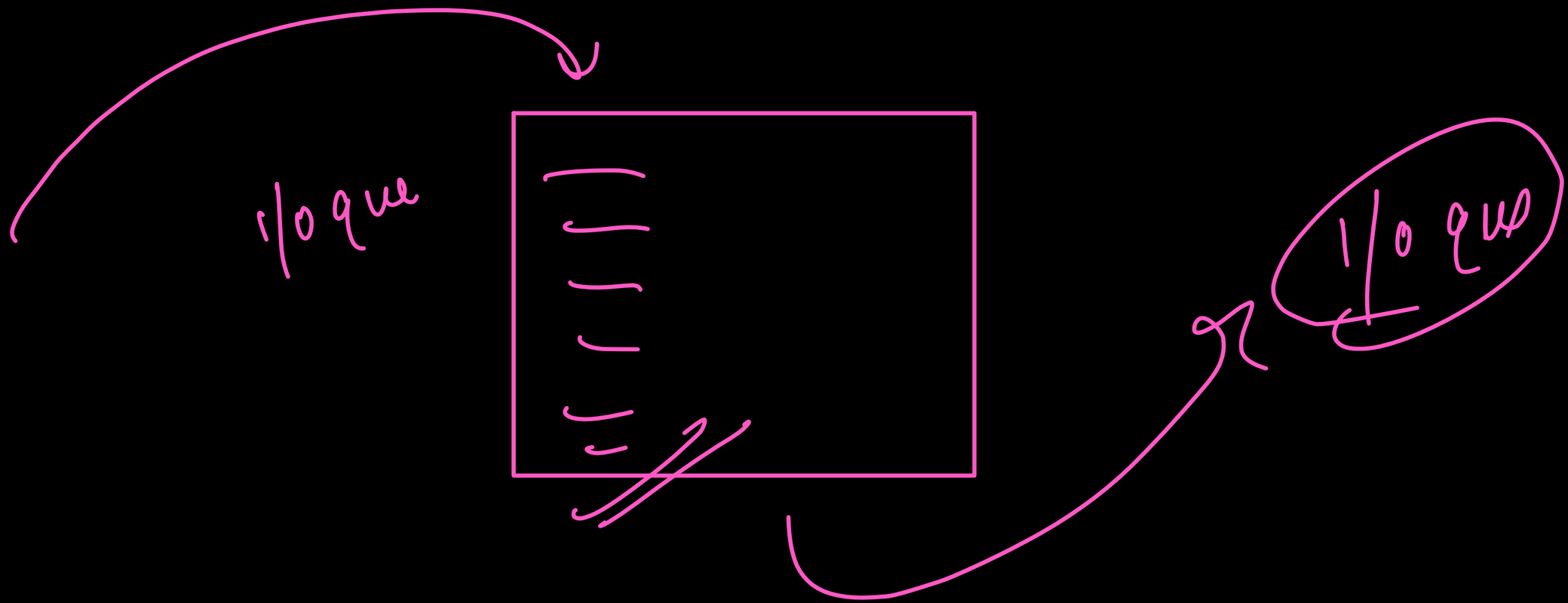


fulfilled
value: undef
onfulfillno: [
]

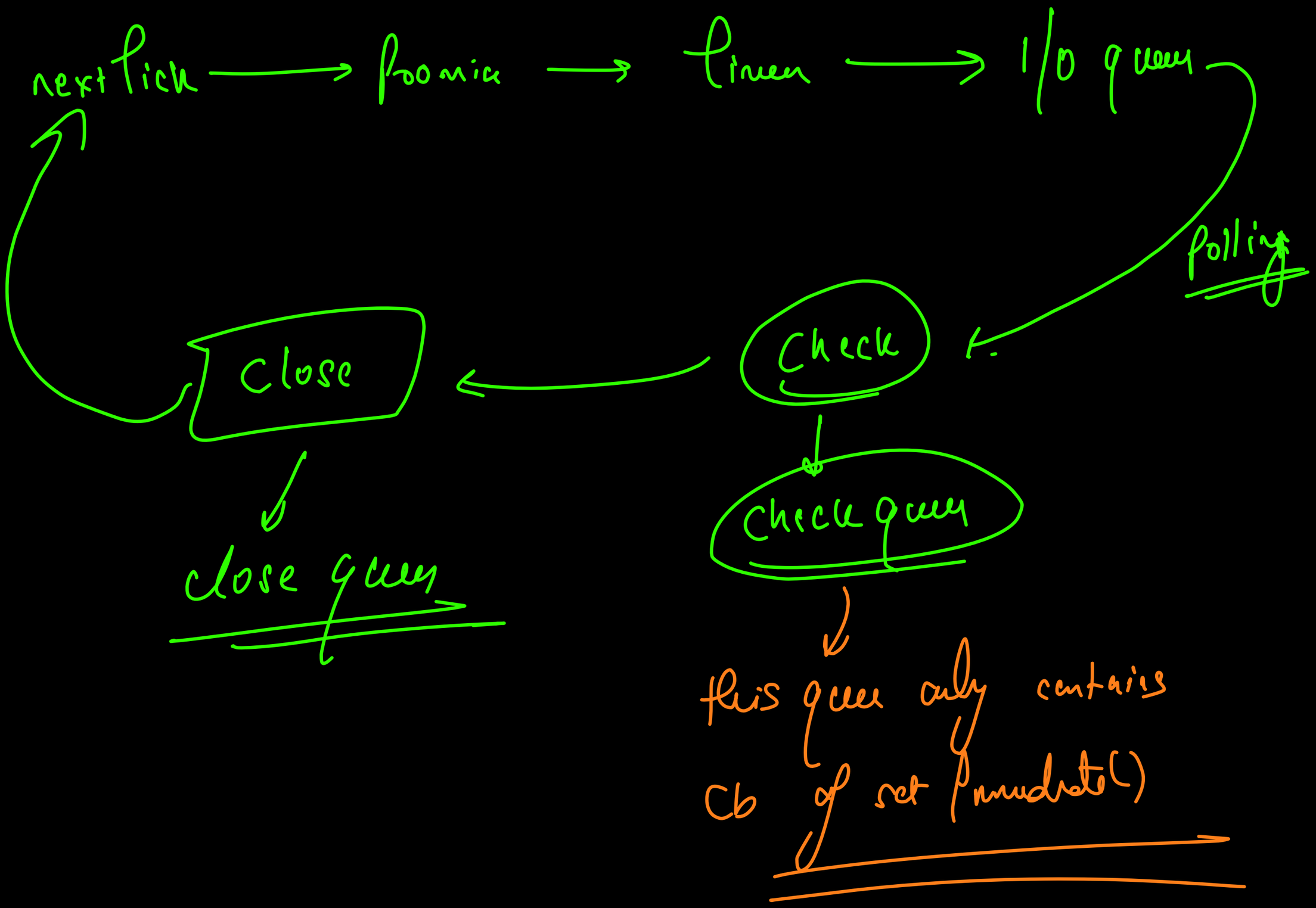
JOIN THE DARKSIDE

1. Any cb in nextTick queue
2. Any cb in Promise queue
3. We one by one execute cb in Timer queue. But after every cb exec, we check step 1 and 2.
4. → Pending callback → I/O queue or callback queue

JOIN THE DARKSIDE



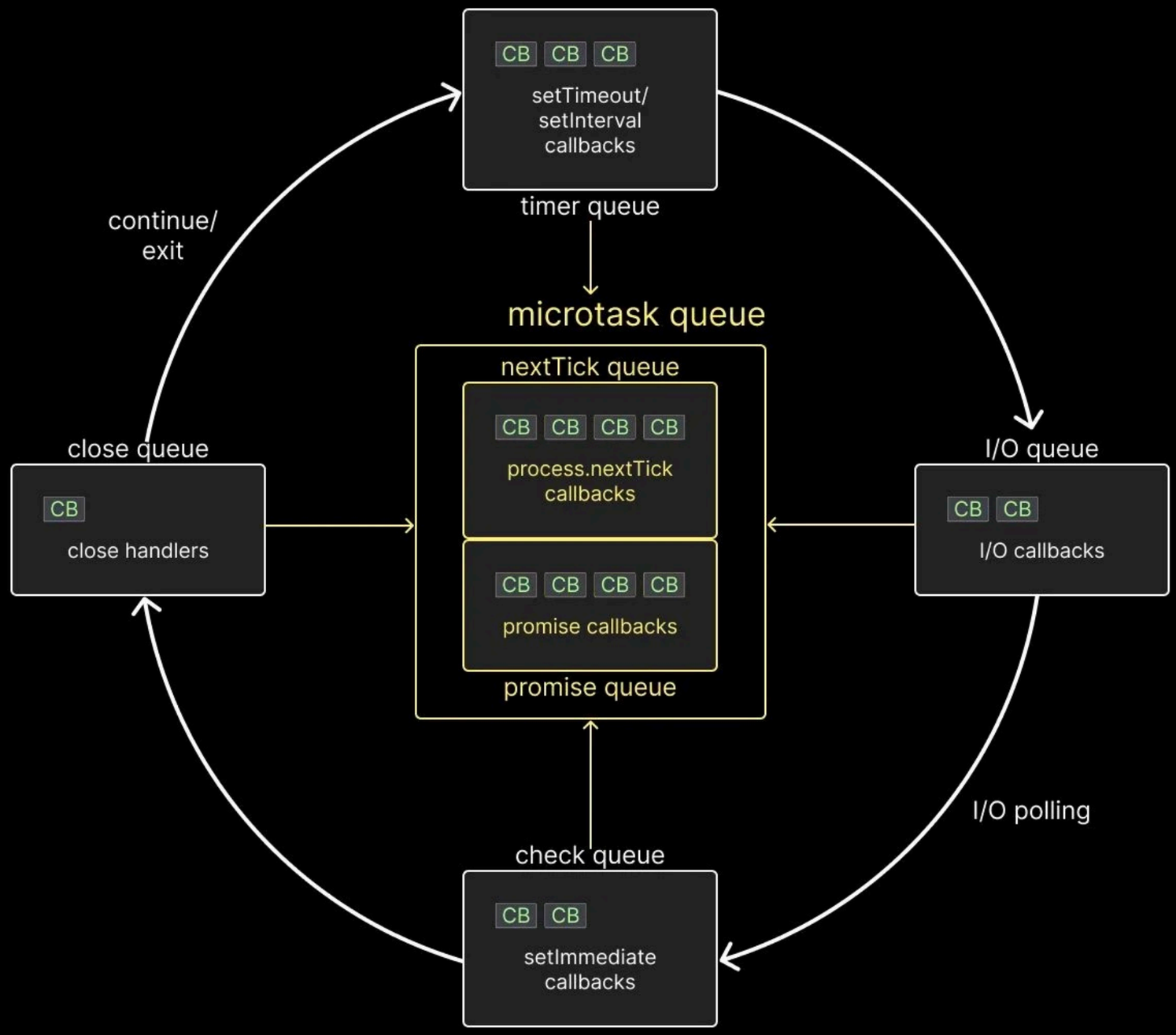
JOIN THE DARKSIDE



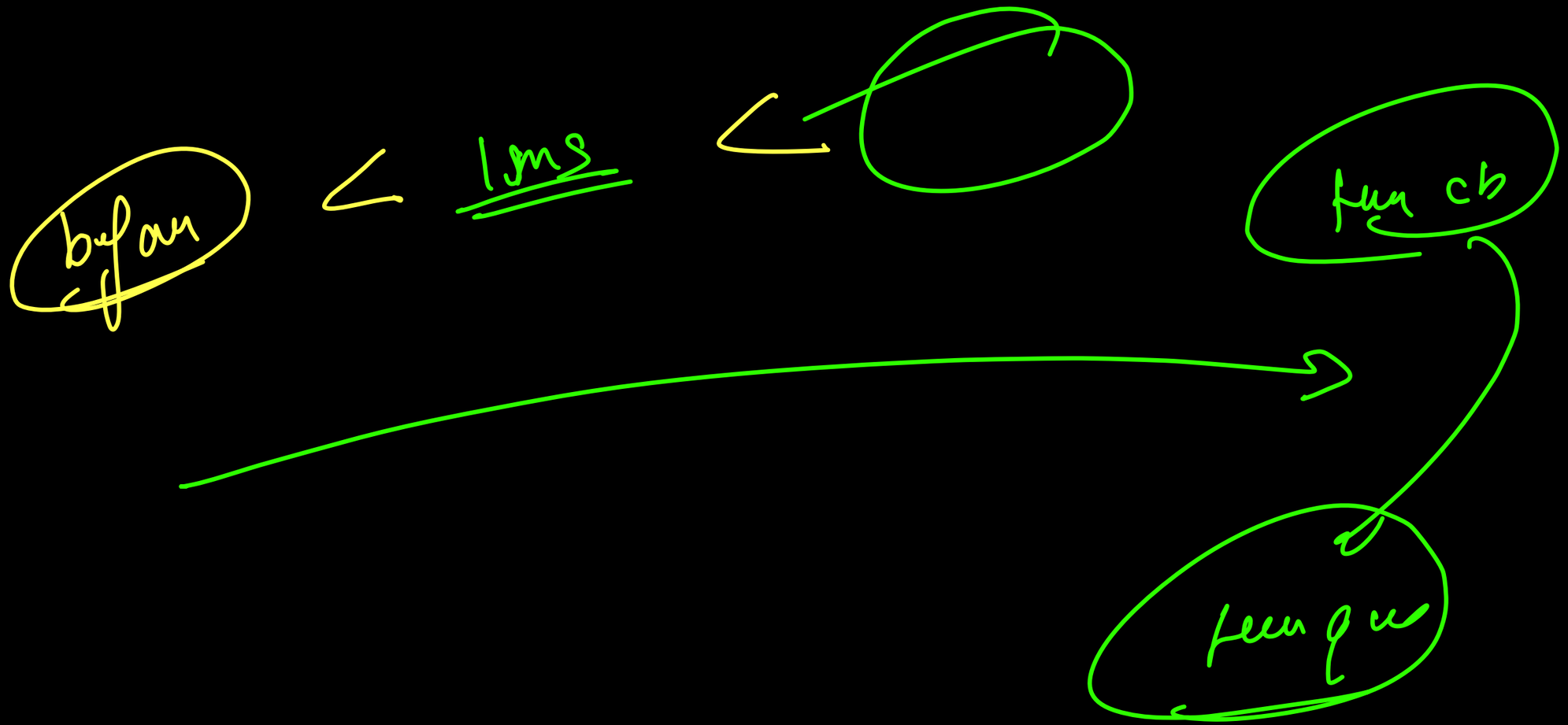
JOIN THE DARKSIDE

```js
const fs = require('fs');

fs.readFile('./readme.md', 'utf8', (err, data) => { // I/O queue - Callback queue
    if (err) {
      console.error(err);
      return;
    }
    console.log(data);
});

process.nextTick(() => console.log("Next tick"));
Promise.resolve(() => console.log("Promise 1"));
setTimeout(() => console.log("Timer 1"), 0);
setImmediate(() => console.log("Immediate callback"));

for(let i = 0 ; i < 10000000000; i++ ) {} // > 5s
```

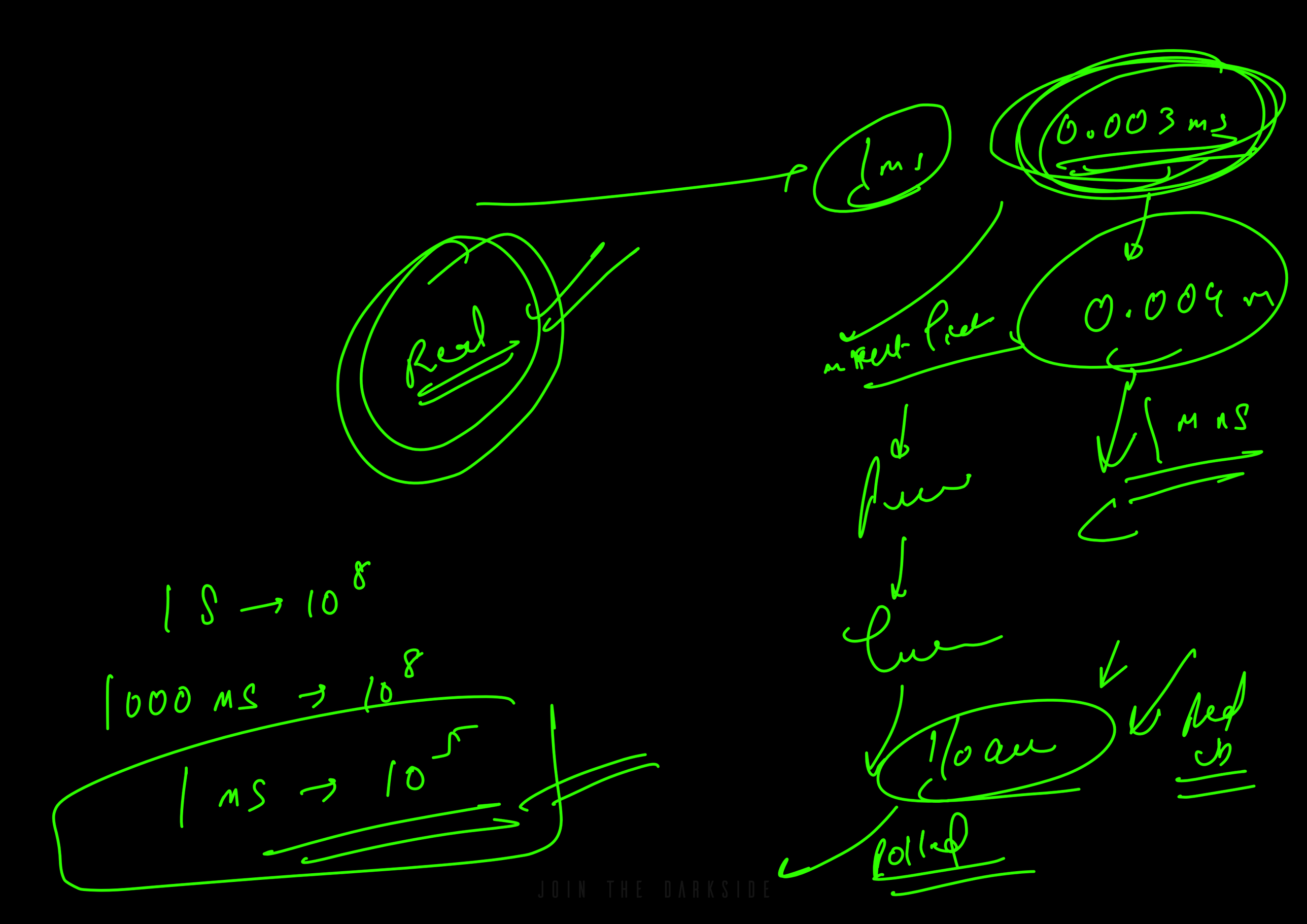
JOIN THE DARKSIDE



JOIN THE DARKSIDE



JOIN THE DARKSIDE



Real

1 ms

0.003 ms

0.004 m

1 ms

Current Pick

Power

Power

1/0 ans

Rolled

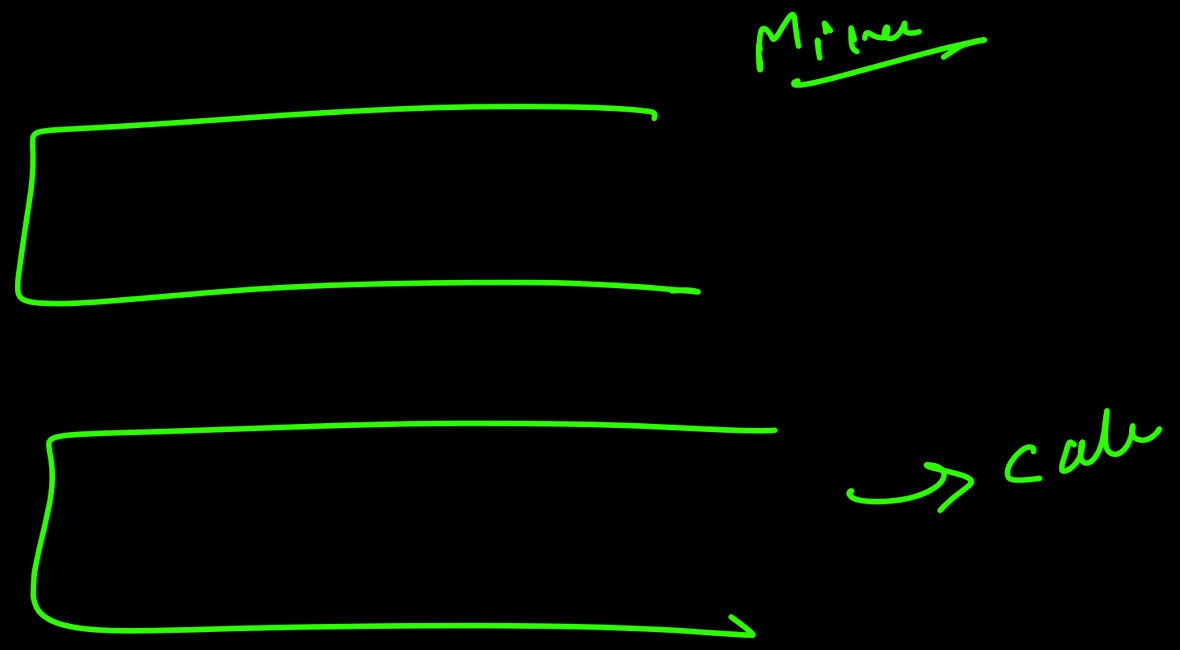
Red ch

$1\,s \rightarrow 10^{8}$

$1000\,ms \rightarrow 10^{8}$

$1\,ms \rightarrow 10^{5}$

JOIN THE DARKSIDE



JOIN THE DARKSIDE