# Work on project. Stage 3/6: Miner mania



Project: Blockchain

Hard 11 minutes

## §1. Description

The blockchain itself shouldn't create new blocks. The blockchain just keeps the chain valid and accepts the new blocks from outside. In the outside world, there are a lot of computers that try to create a new block. All they do is search for a magic number to create a block whose hash starts with some zeros. The first computer to do so is a winner, the blockchain accepts this new block, and then all these computers try to find a magic number for the next block.

There is a special word for this: **mining**. The process of mining blocks is hard work for computers, like the process of mining minerals in real life is hard work. Computers that perform this task are called **miners**.



Note that if there are more miners, the new blocks will be mined faster. But the problem is that we want to create new blocks with a stable frequency. For this reason, the blockchain should regulate the number N: the number of zeros at the start of a hash of the new block. If suddenly there are so many miners that the new block is created in a matter of seconds, the complexity of the next block should be increased by increasing the number N. On the other hand, if there are so few miners that process of creating a new block takes longer than a minute, the number N should be lowered.

In this stage, you should create a lot of threads with miners, and every one of them should contain the same blockchain. The miners should mine new blocks and the blockchain should regulate the number N. The blockchain should check the validity of the incoming block (ensure that the previous hash equals the hash of the last block of the blockchain and the hash of this new block starts with N zeros). At the start, the number N equals 0 and should be increased by 1 / decreased by 1 / stays the same after the creation of the new block based on the time of its creation.

> Do not exit main method until you print 5 blocks! Output is checked right after exiting main method.

## §2. Output example

To be tested successfully, program should output information about first five blocks of the blockchain. Blocks should be separated by an empty line.

```
Block:
Created by miner # 9
Id: 1
Timestamp: 1539866031047
Magic number: 23462876
Hash of the previous block:
0
Hash of the block:
1d12cbbb5bfa278734285d261051f5484807120032cf6adcca5b9a3dbf0e7bb3
Block was generating for 0 seconds
N was increased to 1

Block:
Created by miner # 7
Id: 2
Timestamp: 1539866031062
Magic number: 63576287
Hash of the previous block:
1d12cbbb5bfa278734285d261051f5484807120032cf6adcca5b9a3dbf0e7bb3
Hash of the block:
04a6735424357bf9af5a1467f8335e9427af714c0fb138595226d53beca5a05e
Block was generating for 0 seconds
N was increased to 2

Block:
Created by miner # 1
Id: 3
Timestamp: 1539866031063
Magic number: 57875299
Hash of the previous block:
04a6735424357bf9af5a1467f8335e9427af714c0fb138595226d53beca5a05e
Hash of the block:
0061924d48d5ce30e97cfc4297f3a40bc94dfac6af42d7bf366d236007c0b9d3
Block was generating for 0 seconds
N was increased to 3

Block:
Created by miner # 2
Id: 4
Timestamp: 1539866256729
Magic number: 23468237
Hash of the previous block:
0061924d48d5ce30e97cfc4297f3a40bc94dfac6af42d7bf366d236007c0b9d3
Hash of the block:
000856a20d767fbbc38e0569354400c1750381100984a09a5d8b1cdf09b0bab6
Block was generating for 5 seconds
N was increased to 4

Block:
Created by miner # 9
Id: 5
Timestamp: 1539866256749
Magic number: 18748749
Hash of the previous block:
000856a20d767fbbc38e0569354400c1750381100984a09a5d8b1cdf09b0bab6
Hash of the block:
000031e22049646ca25c5f63fcc070e8c76319a050a7d1d5ca402090a30e9612
Block was generating for 15 seconds
N stays the same

Block:
Created by miner # 5
Id: 6
Timestamp: 1539866256750
Magic number: 23423458
Hash of the previous block:
000031e22049646ca25c5f63fcc070e8c76319a050a7d1d5ca402090a30e9612
Hash of the block:
0000e3dc2b8fc5f0c635358aa19a84eae68c316a40d22d6283ab1152f486f003
Block was generating for 65 seconds
N was decreased by 1
```

Code Editor IDE

Solve in IDE

✓ **IDE is responding** IntelliJ IDEA 2019.3

✓ **Plugin is responding** 3.2-2019.3-3686

This content was created 11 months ago and updated 2 days ago. Share your feedback below in comments to help us improve it!

Comments (7) Hints (0) Useful links (0) Solutions (0)

**Share something, Sergey Kubatko**

Post Please do not post solutions here

Sort by: **Last posted**

NA **Nabil AIT SAID** 2 months ago Report

The main Thread shouldn't terminate before the other threads :)

♡ 1 Reply

NA **Nabil AIT SAID** 2 months ago Report

"You should output 5 blocks, found 0"
But everything works well in local.

♡ 0 Reply

CH **Christian H** 4 months ago Report

I don't know if I understand what the problem is asking. If anyone could clarify that would be great. Are all of threads intended to be working on the same block? Or am I passing a block to the executor where it'll decide on which "miner" gets it?

♡ 0 Reply

WL **William Liu** 6 months ago Report

Getting:

Wrong answer in test #1 Every block should contain 9 lines of data

The stdout is identical to the example...

♡ 0 Reply

TN **Toni Nagy** 6 months ago Report

So I found the problem, you need to output the whole blockchain in the main thread and not in other threads. So you shouldn't output a successfully mined block as soon as it was found by a random thread, but wait for them to finish and then output the whole thing in main. I put a Thread.sleep(5000) method for this, which should be more than enough considering you should only test it for the first five blocks (which my system found typically in maximum one second).

♡ 0 Reply

TN **Toni Nagy** 6 months ago Report

My problem is the same as Khiladi's,
"You should output 5 blocks, found 0"
But everything seems to work fine.

♡ 0 Reply

K **Khiladi** 7 months ago Report

Hi, I am getting a failed test case saying "You should output 5 blocks, found 0". Interestingly, the same code works fine in my local Intellij setup producing the same output as above. Any help will be appreciated, thanks

♡ 1 Reply