Blockchain Foundations **NTUA, Sping 2026**

Problem Set #6 *

Soft Deadline: TBD
Final Deadline: TBD

1 Mining Blocks

In this exercise, you will (finally) mine your own blocks.

1. Collect transactions in your mempool into a block.
2. Create the logic to build a block template (which must contain the student IDs of your team members).
3. Mine on the candidate block to discover proof-of-work.
4. Benchmark and optimize your miner. What hashrate can you achieve?
5. Broadcast your newly discovered block when it is found.
6. It is in the spirit of blockchains to make everything publicly verifiable and open source. As part of this process, please now *make your code open source*. You can do this after the teaching staff has confirmed on Ed that all the programming assignments have been graded. Choose an open source license (such as AGPL, GPL, MIT, or BSD) and edit your GitHub repository settings to make your repository public. In addition to sharing your code with fellow students, this will help showcase your experience as a blockchain professional. You can later revise your code and clean it up or add additional features.
7. Optional: In the event of receiving conflicting transactions, optimize the set of transactions that you include in the block to maximize the fees you can collect. (Note that there is no block size limit in Marabu. So if there are no double spending transactions, you are expected to include all valid transactions that you have seen.)
8. Optional: In order to drop your lowest scoring homework, create a transaction that pays 50 bu to the following address, and add it to your Gradescope submission:

   `3f0bc71a375b574e4bda3ddf502fe1afd99aa020bf6049adfe525d9ad18ff33f`

*Version: 2 – Last update: July 1

The transaction must have a valid signature, spend rightfully created money, be broadcast to the network and make it to the longest chain. We will consider the payment received when the transaction is confirmed into a block that is buried under $k = 6$ blocks. Make sure you broadcast your transaction early so that it is included in a block before the hard deadline written above. It must also be money sent from a public key for which you control the private key (to prove that it is you). For this part, we don't care if you mined the money or if you received it from someone else.

2 Sample Test Cases

- You get full points for this homework if the longest valid chain contains a block mined by your node.
- The block must be k-deep with $k = 6$ in the longest valid chain (as seen by the grader) at the time of the hard deadline, in order to be accepted.
- The block must contain one of the transactions signed by the public key

  `3f0bc71a375b574e4bda3ddf502fe1afd99aa020bf6049adfe525d9ad18ff33f`

  Transactions signed by this key will be broadcast to the network at regular intervals.
- The block must contain a `studentids` field with the SUNet ids (the string, not the 8-digit number) of your team members in the format shown below:

  `"studentids": ["id1", "id2", "id3"]`
- There will be no grading right after the soft deadline. However, we recommend that you finish your implementation before the soft deadline so that you have enough time to mine a block before the hard deadline.
- For the hard deadline, fill the Gradescope as usual, but also enter the blockid of the block that you have mined.
- Attacks might be happening on the network[1]. Once you build it, keep running your miner together with your fellow students to ensure the honest majority property holds. This will ensure the honest chain survives as the longest chain and there are no violations of chain growth, chain quality, or common prefix, which could be detrimental to the outcome of this homework!

If you're using JavaScript or TypeScript, you may need to use *workers* to allow your miner to run in parallel with your full node. See `https://nodejs.org/api/worker_threads.html` for more details.

[1] Totally not by us.