# Level 6 - Blockchain

Definitions

- Funds are passed on through transaction linking
- Transactions need to be included in blocks to be booked
- Transaction requests need to be fulfilled
- New Blocks need to be filled with transactions from transaction requests

Task

- Report the transactions in the longest chain of valid blocks
  - If there are multiple, the one that reached their last block earlier
- Report the longest chain of blocks

Level 6

Chain Example - unsolved
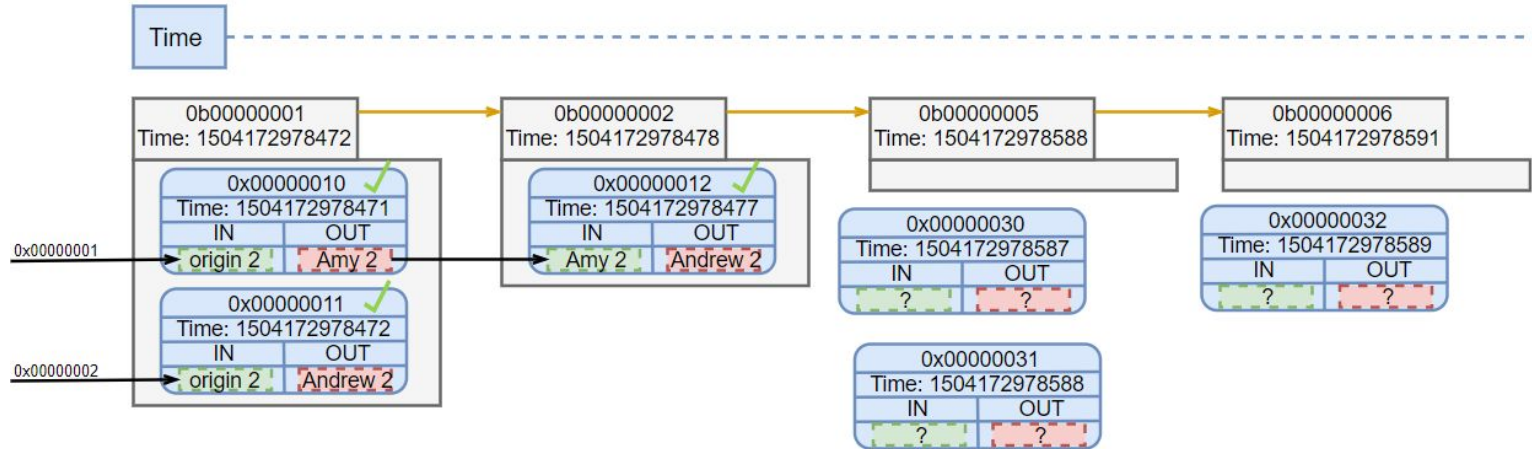
Filling Blocks

Transaction Requests

- New blocks need to be filled with **valid** transactions generated **from the transaction requests only**.

Filling Blocks

- New blocks need to be filled with **valid** transactions generated **from the transaction requests only**.
    - Criteria for processing a transaction request as defined in level 4 **for the respective branch of the blockchain**
- Blocks are filled up to their maximum capacity with transaction requests that are available at latest at block generation time.
    - Transactions are selected in chronological order
- Transaction requests that can't be put in a block because it is already filled are queued for the next block on that branch

Data format

Input (same as in level 5)

`<NumberOfTransactions>` the number of transactions in the banking system

NumberOfTransactions lines: `<TransactionID> <NumberOfInputs>` NumberOfInputs * InputElement `<NumberOfOutputs>`

NumberOfOutputs * OutputElement `<TransactionSubmitTime>`

InputElement: `<InputTransactionID> <InputTransactionOwner> <InputTransactionAmount>`

OutputElement: `<OutputTransactionOwner> <OutputTransactionAmount>`

`<NumberOfBlocks>` the number of blocks in the banking system

NumberOfBlocks lines: `<BlockId> <PreviousBlockId> <NumberOfTransactions>` NumberOfTransactions * `<TransactionId> <BlockCreationTime>`

Additional Input

`<NumberOfTransactionRequests>` the number of transaction requests

NumberOfTransactionRequests lines: `<TransactionID> <FromOwner> <ToOwner> <Amount> <TransactionSubmitTime>`

`<NumberOfBlocksToFill>` the number of blocks which have to be filled with the new transactions

NumberOfBlocksToFill lines: `<BlockId> <PreviousBlockId> <BlockCreationTime>`

Level 6

Data format

Output (same as in level 5)

`<NumberOfTransactions>` the number of transactions in the longest valid blockchain

NumberOfTransactions lines: `<TransactionID> <NumberOfInputs>` NumberOfInputs * InputElement `<NumberOfOutputs>` NumberOfOutputs * OutputElement `<TransactionSubmitTime>`

InputElement: `<InputTransactionID> <InputTransactionOwner> <InputTransactionAmount>`

OutputElement: `<OutputTransactionOwner> <OutputTransactionAmount>`

`<NumberOfBlocks>` the number of blocks in the longest valid blockchain

NumberOfBlocks lines: `<BlockId> <PreviousBlockId> <NumberOfTransactions>` NumberOfTransactions * `<TransactionId> <BlockCreationTime>`

Level 6

Example

Input

```
3
0x00000010 1 0x00000001 origin 2 1 Amy 2 1504172978471
0x00000011 1 0x00000002 origin 2 1 Andrew 2 1504172978472
0x00000012 1 0x00000010 Amy 2 1 Andrew 2 1504172978477
2
0b00000001 0b00000000 2 0x00000010 0x00000011 1504172978472
0b00000002 0b00000001 1 0x00000012 1504172978478
3
0x00000030 Andrew Amy 3 1504172978587
0x00000031 Andrew Amy 3 1504172978588
0x00000032 Andrew Amy 1 1504172978589
2
0b00000005 0b00000002 1504172978588
0b00000006 0b00000005 1504172978591
```

Output

```
5
0x00000010 1 0x00000001 origin 2 1 Amy 2 1504172978471
0x00000011 1 0x00000002 origin 2 1 Andrew 2 1504172978472
0x00000012 1 0x00000010 Amy 2 1 Andrew 2 1504172978477
0x00000030 2 0x00000011 Andrew 2 0x00000012 Andrew 2 2 Amy 3 Andrew 1 1504172978587
0x00000032 1 0x00000030 Andrew 1 1 Amy 1 1504172978589
4
0b00000001 0b00000000 2 0x00000010 0x00000011 1504172978472
0b00000002 0b00000001 1 0x00000012 1504172978478
0b00000005 0b00000002 1 0x00000030 1504172978588
0b00000006 0b00000005 1 0x00000032 1504172978591
```

Level 6

Chain Example - solved