This homework exam consists of 3 questions. You can earn a total of 100 points. You may discuss questions and approaches on a high level with your fellow students, but must write down your own solutions. The first 10 points can be obtained by following these rules:

- Hand in your solutions on paper.
- Write clearly and legibly, or use a type setting system (such as LATEX).
- Write your name and student number on each sheet you hand in, and use page numbers.

Make sure you understand the questions, and be concise in your answers. An explanation of how you obtained your answer is not mandatory unless specified in the question; however, a short derivation may be helpful in case of arithmetic mistakes.

**Question 1**

In a Lotto drawing, six balls are drawn randomly from a set of 44 balls with integer numbers on them. One of these balls is gold, the other 43 are green. I am a big fan of Lotto, and I watched the last 1000 drawings on TV and took notes of the results. I noticed that in all those drawings, the gold ball was drawn 119 times, which is less often than it statistically should be drawn. I also noticed that the game host always tends to rub his fingers together in peculiar way after drawing the gold ball, and now I suspect that the paint of the gold ball is stickier than the paint on the green balls, and this is why the gold ball is drawn less often than it should! Luckily, I have a degree in mathematics, and rather than write an angry letter to the TV channel hosting the Lotto right away, I decide to first verify my suspicion.

(a) Formulate a hypothesis based on my suspicion.

(b) Test the hypothesis, using a confidence of 0.05.

**Question 2**

A red-black tree is a data structure to store a set of numbers. It is an alternative to AVL trees as a way to keep a binary tree balanced. In a red-black tree, every node gets assigned a colour (either red or black), according to the following set of rules:

- The root is black.
- A red node never has a red child.
- Every path from the root of the tree to any node which has at most one child contains the same number of black nodes.

(a) Prove that the height of any red-black tree with \( n \) nodes is always \( \Theta(\log n) \).

When inserting nodes into a red-black tree, we have to make sure that the properties are maintained. We first insert the node as a new leaf (in the proper location) and paint it red. Now, there could be a red node \( v \) with a red parent. While this is the case, consider the (black) grandparent \( w \) of \( v \). If \( w \) has two red children, we can paint \( w \) red and its children black, and all is well except that now \( w \) could be a red node with a red parent; we recurse higher up the tree. Otherwise, we can perform a rotation on \( w \). Finally, if, after the insertion, the root has become red, we simply paint it black - note that this will never violate the other rules.

Consider the following sequence of numbers: \( S = \{1, 2, 3, 6, 5, 4\} \).

(b) Draw both the AVL tree and the red-black tree we would get when inserting the numbers in \( S \) into an initially empty tree.

(c) What are the advantages and disadvantages of using a AVL tree versus a red-black tree?
Crarstaft is a symmetric 2-player simultaneous game in which players can choose to make one of three moves: zinglerg (Z), hisklydra (H), or miskluta (M). The game has this payoff table:

<table>
<thead>
<tr>
<th></th>
<th>Z</th>
<th>H</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>(0,0)</td>
<td>(50,0)</td>
<td>(0,200)</td>
</tr>
<tr>
<td>H</td>
<td>(0,50)</td>
<td>(0,0)</td>
<td>(100,0)</td>
</tr>
<tr>
<td>M</td>
<td>(200,0)</td>
<td>(0,100)</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

(a) List all (pure and mixed) equilibria of Crarstaft.

In *iterated Crarstaft*, two players play a sequence of 10 rounds of Crarstaft in a row. Each round, they can decide to change their move. The game ships with three AIs:

- **AI A**: Always plays M.
- **AI B**: Starts with H, and then always plays what its opponent played in the previous round.
- **AI C**: Starts with Z, and then always plays what would beat what its opponent played in the previous round. So, if its opponent played Z before, AI C will play M now; if its opponent played M, AI C will play H, and if its opponent played H, AI C will play Z.

(b) We let the three AIs play iterated Crarstaft against each other in 6 different pairings. What are the outcomes?

(c) Can you design a (deterministic) new AI that will beat all three AIs?

After players complained that the AIs in Crarstaft were too easy to beat, the long-awaited expansion of Crarstaft ships with a fourth, randomized, AI.

- **AI D**: Always plays a random move, according to the mixed equilibrium found in (a).

(d) What is the expected outcome of AI D when playing against each other AI? Also answer the question for the AI you designed yourself in (c).

(e) Can you design a (randomized) new AI that is expected to beat AIs A, B, C and D?