• Exercises appear at the i1 homepage (http://algo.rwth-aachen.de/en/Lehre/SS17/Online.php) on Friday.
• You have seven days to create a solution and it must be done in a group of two or three students.
• Write the name, group number and enrollment number of each group member on every sheet that you hand in.
• To achieve the permission for the exam you must earn 50% of the sum of all points and present one of your solutions at least once.
• You can earn 50% bonus points by presenting your solution. At the beginning of every exercise session, you can mark the exercises that you want to present.
• If a student is not able to present a correct solution although he/she marked the exercise as presentable, he/she will lose all of his/her points on the exercise sheet.

Exercise 1 (2 points)
Proof that flush when full ($FWF$) is also a marking algorithm.

Exercise 2 (2 points)
Think of the following modification of the paging problem, we call it $\Delta^-$-paging. The additional input parameter $\Delta$ has the following effect: If page $p_i$ is requested in step $l$ ($x_l = p_i$), the adversary can NOT ask for the pages $\{p_i-\Delta, \ldots, p_i+\Delta\}$ in step $l + 1$. Thus, $x_{l+1} \notin \{p_i-\Delta, \ldots, p_i+\Delta\}$.

Show that it is possible to create an instance $I$ with $m + \Delta + 1$ different pages from an arbitrary instance for the normal paging problem $I'$ with $m$ different pages such that for three algorithms the competitive ratios still hold. Give also two algorithms for which the transformation will not work and explain briefly why.

Exercise 3 (4 points)
Think of the following modification of the paging problem, we call it $\Delta^+$-paging. The additional input parameter $\Delta$ has now the following effect: If page $p_i$ is requested in step $l$ ($x_l = p_i$), the adversary can ONLY ask for the pages $\{p_i-\Delta, \ldots, p_i+\Delta\}$ in step $l + 1$. Thus, $x_{l+1} \in \{p_i-\Delta, \ldots, p_i+\Delta\}$.

Does this restriction for the adversary help the algorithm? Proof your answer.

Exercise 4 (4 points)
Think of the following modification of the paging problem, we call it 2List-paging. The adversary has two Lists $L_1$ and $L_2$ of pages and corresponding pointers $l_1$ and $l_2$. One is sorted in ascending order and the other one in in descending order. Thus $L_1 = [p_{k+1}, \ldots, p_m, p_1, \ldots, p_k]$ and $L_2 = [p_m, \ldots, p_1]$. The adversary can ask for a page that are on the pointer positions. They are initialized with 0 and increase if the corresponding List is used. So, in the first step, the adversary can only ask for $p_{k+1}$ or $p_m$. If $p_{k+1}$ is requested, $l_1$ is increased by one and the adversary can ask for page $p_{k+2}$ or $p_m$. If a pointer reaches the end of the List, it is set to 0 and starts from the beginning.

Does this modification help the algorithm? Proof your answer.

Exercise 5 (4 points)
A hungry cow stands in front of the farmers fence. She knows that if she follows the fence in one direction she will reach the barn where she can eat. But she does not know the correct direction.

As a cow, she does the obvious: she walks one cow meter to one direction, then back and 2 cow meters into the other direction, then back and so on. Thus, the cow walks in step $i \in \{0, \ldots\} 2^i$ steps into one direction measured from the start. Proof that this approach cannot have a competitive ratio smaller than 9.