Exercise 1 (3 points)

The paging problem is an online minimization problem. Let there be a positive integer $k$ which denotes the cache size. Initially the cache is filled with the first $k$ pages $B_0 = \{p_1 \ldots p_k\}$. There are $m$ different pages. An input sequence has the form $I = \{x_1, x_2, \ldots, x_n\}$ with $x_i \in \{p_1, p_2, \ldots, p_m\}$ as the page that arrives in time step $i$. If a page $x_i$ in time step $i$ is requested and the page is not part of the current cache ($x_i \notin B_{i-1}$), the algorithm has to replace a page $p_j \in B_{i-1}$ such that the new cache is of the form $B_i = (B_{i-1} \setminus \{p_j\}) \cup \{x_i\}$. If a page is replaced in step $i$, the output is $y_i = 1$. Otherwise the output is $y_i = 0$.

The goal is to minimize the cost of an algorithm $A$ which is defined as $cost(A) = \sum_{i=1}^{n} y_i$.

a) Assume you get the paging problem as offline problem, so $I$ is known beforehand. Describe in a few sentences how your algorithm would find the optimal solution. We denote the cost of an optimal solution as $OPT$.

b) You have three different types of algorithms. A Stack $A_1$, a Queue $A_2$ and an algorithm $A_3$ that replaces the page which was least frequently used. Derive for every algorithm an instance of the paging problem with $m \geq 5$, such that $cost(A_i) \geq k \cdot OPT$ holds.

Exercise 2 (3 points)

A friend asks you for help. He/She wants to go on a ski trip. Both of you do not know how the weather will be. But there are only two states: good enough for skiing or not. Your friend is for $n$ days on the trip and you can decide each day to rent or buy your ski with the knowledge of the weather for the current day. Renting costs 1 and buying $k$. If skis are bought they will not be rented anymore.

Can you tell your friend what to do? Proof that your approach is never worse than $m$ times the cost that would appear if you would know the weather beforehand. Minimize the $m$ (You do not need to prove that your $m$ is minimal).