

Übungsblatt 1

Aufgabe 1.1

Eine hungrige Kuh ist auf der Suche nach einer Weide. Sie wandert ziellos durch die Gegend, bis sie auf einen Pfad trifft, von dem sie weiß, dass er zu einer Weide führt. Leider hat sie vergessen, in welche Richtung des Pfades sie wandern muss, um zu der Weide zu gelangen. Sie weiß lediglich, dass die Weide mindestens eine Längeneinheit entfernt ist.

- (a) Geben Sie einen deterministischen Online-Algorithmus mit strikt kompetitivem Faktor 9 an, mit dem die Kuh die Weide finden kann.
- (b) Zeigen Sie, dass für $r < 3$ kein deterministischer strikt r -kompetitiver Online-Algorithmus für dieses Problem existiert.
- (c) Zeigen Sie, dass für $r < 9$ kein deterministischer strikt r -kompetitiver Online-Algorithmus für dieses Problem existiert.
- (d) Geben Sie einen randomisierten Algorithmus an, der strikt 7-kompetitiv ist.

Aufgabe 1.2

Der Paging-Algorithmus CLOCK verwaltet die Cache-Seiten in einer zyklischen *Queue*. Jede einzelne ist mit einem *Used-Bit* ausgestattet. Es gibt einen Zähler, der auf die letzte Seite zeigt, die in den Cache geladen wurde. Sobald auf eine Seite zugegriffen wird, wird ihr Used-Bit auf 1 gesetzt. Bei einem Seitenfehler wird der Zähler um eins erhöht und der Algorithmus sucht ab dieser Stelle eine Seite, deren Used-Bit 0 ist. Die Used-Bits aller bis dahin betrachteten Seiten werden auf 0 gesetzt.

- (a) Zeigen Sie, dass CLOCK kein Markierungsalgorithmus ist.
- (b) Zeigen Sie, dass CLOCK k -kompetitiv ist.

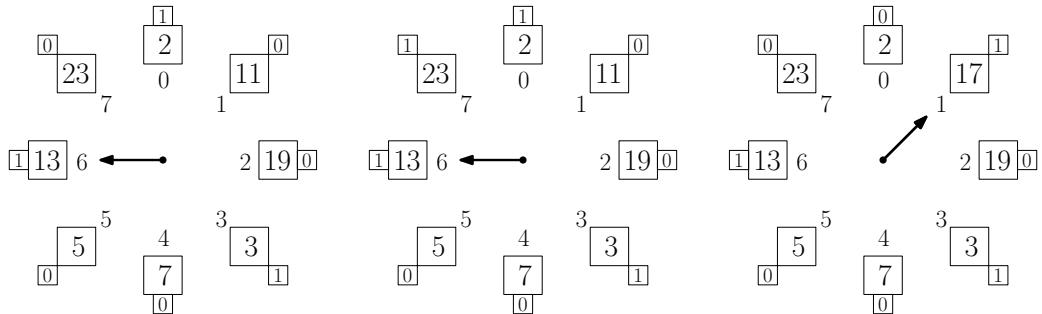


Abbildung 1: Veranschaulichung des CLOCK-Algorithmus; *links*: Situation zu einem willkürlichen Zeitpunkt - Seite 13 wurde als letzte in den Cache geladen; *Mitte*: Zugriff auf Seite 23, die bereits im Cache ist - Used-Bit wird auf 1 gesetzt; *rechts*: Zugriff auf Seite 17, also Seitenfehler - der Zähler wird von 6 über 7 und 0 auf 1 bewegt, wo das Used-Bit das erste Mal 0 ist.

Aufgabe 1.3

Bei **BIN PACKING** sind n Objekte mit Gewichten $w_1, \dots, w_n \in [0, 1]$ und eine unbegrenzte Anzahl an Eimern (*Bins*) mit Kapazität 1 gegeben. Ziel ist es, die n Objekte in so wenig wie möglich Eimer zu packen. Formal ausgedrückt suchen wir die kleinste Zahl k , für die die Menge $\{1, \dots, n\}$ in k Klassen S_1, \dots, S_k partitioniert werden kann, sodass für jeden Index $i \in \{1, \dots, k\}$ die Ungleichung $\sum_{j \in S_i} w_j \leq 1$ erfüllt ist. Dieses Problem ist \mathcal{NP} -schwer.

Bei der Online-Variante von **BIN PACKING** muss jedes Objekt i einem Eimer zugeteilt werden, ohne die Anzahl n der Objekte oder die Gewichte der Objekte $i + 1, \dots, n$ zu kennen.

Geben Sie einen strikt 2-kompetitiven Online-Algorithmus für **BIN PACKING** an.

Aufgabe 1.4

Beweisen Sie, dass kein deterministischer Online-Algorithmus für das **BIN PACKING**-Problem besser als $4/3$ -kompetitiv ist.

Hinweis: Betrachten Sie die Sequenz σ der Länge $2m$ für beliebiges $m \in \mathbb{N}$. Die ersten m Einträge der Sequenz betragen $1/2 - \epsilon$. Die letzten m Einträge sind gegeben durch $1/2 + \epsilon$. Betrachte Sie zunächst die verkürzte Sequenz σ' bestehend aus den ersten m Einträgen von σ . Nehmen Sie an, dass der Online-Algorithmus hier b Eimer befüllt.