Thomas Kesselheim 19. Oktober 2017
Abgabe: in der Vorlesung am 26. Oktober 2017

Algorithmen und Unsicherheit

Wintersemester 2017/18

Übungsblatt 2

Aufgabe 1: (5 Punkte)
Betrachte den Paging-Algorithmus Flush When Full (FWF). Wann immer der Cache voll ist und eine Seite verdrängt werden müsste, lösche alle Seiten im Cache. Zeige, dass dieser Algorithmus k-competitive ist.

Aufgabe 2: (5 Punkte)
Betrachte den Paging-Algorithmus LIFO: Verdränge immer die zuletzt hinzugefügte Seite. Zeige, dass dieser Algorithmus nicht c-competitive ist für alle c, die nicht von der Sequenzlänge abhängen.

Aufgabe 3: (5 Punkte)
Nun betrachten wir den LRU-Algorithmus aus der Vorlesung, vergleichen ihn aber mit dem Offline-Optimum OPT', das nur eine Cache-Größe von $h < k$ hat. Zeige, dass es ein b gibt, so dass für jede Sequenz σ gilt

$$\mathbf{E}\left[\mathrm{cost}(\mathrm{ALG}(\sigma))\right] \leq \frac{k}{k-h+1}\mathrm{cost}(\mathrm{OPT}'(\sigma)) \ .$$

Hinweis: Die meisten Teile der Analyse aus der Vorlesung können weiter verwendet werden, müssen also auch nicht neu bewiesen werden.

Aufgabe 4: (5 Punkte)
Wir betrachten folgendes Problem: Wir möchten Ski fahren gehen, wissen aber nicht an wie vielen Tagen. Modelliert wird dies durch eine Anfragesequenz mit $\sigma_t = 1$, wenn wir am t. Tag Skifahren möchten, sonst $\sigma_t = 0$.
Wir können Skier leihen (Kosten 1 pro Tag) oder kaufen (Kosten B einmalig). Zu Beginn eines Skitages entscheiden wir uns, ob wir Skier für diesen Tag leihen oder kaufen wollen, sofern wir noch keine Skier gekauft haben.

Zeige, dass folgender Algorithmus 2-competitive ist: Leihe Skier für bis zu B Tage, am $B+1$. Tag kaufe Skier.