
3. Übungsblatt

— Lösungen —

Aufgabe 4 (Stapelproblem). (2 + 1 Punkte)

Wir betrachten das folgende Stapelproblem: Eine Menge $N := \{1, \dots, n\}$ von Elementen soll *konfliktfrei* gestapelt werden, so dass die Anzahl der benötigten Stapelplätze minimiert wird. Eine Anfragesequenz $\sigma = (r_1, \dots, r_n)$ ist eine beliebige Permutation von N , die angibt, in welcher Reihenfolge die Elemente gestapelt werden müssen. Bei einer Anfrage $r_i \in N$ legt der online Algorithmus ALG das Element r_i entweder auf einen bereits verwendeten oder auf einen neuen Stapelplatz ab. Wir nehmen an, dass wir nur von oben auf die Stapel zugreifen können; r_i wird also das neue oberste Element des gewählten Stapels. Zwei Elemente $i, j \in N, i \neq j$, erzeugen einen *Konflikt* genau dann, wenn das Element j über i liegt und $i < j$. Das Ziel ist es, die n Elemente konfliktfrei zu stapeln und dabei die Anzahl der benötigten Stapelplätze zu minimieren.

- (a) Entwickle einen online Algorithmus, der die Anzahl der benötigten Stapelplätze minimiert.

(Hinweis: Eine Teilfolge $\sigma' = (i_1, i_2, \dots, i_\ell)$ von σ heißt *steigend* wenn $i_1 < i_2 < \dots < i_\ell$ gilt. Überlege, was die Länge einer steigenden Teilfolge von σ mit den Anzahl der benötigten Stapelplätze einer optimalen Lösung zu tun hat.)

- (b) Nimm an, dass wir auf jeden Stapelplatz maximal h Elemente stapeln können. Entwickle einen $(2 - 1/h)$ -kompetitiven online Algorithmus für dieses Problem.

Lösung:

Zu (a): *Der Algorithmus betrachtet die bereits verwendeten Stapel immer in einer festen Ordnung (von links nach rechts) und legt das Element r_i auf den ersten Stapelplatz ab, auf dem es keinen Konflikt verursacht. Gibt es keinen solchen Stapel, öffnet ALG einen neuen Stapelplatz, fügt diesen ans Ende der Ordnung ein (weitesten rechts) und legt r_i darauf ab. Die Zulässigkeit der konstruierten Lösung ist damit gegeben.*

Die Länge der längsten steigenden Teilfolge von σ ist eine untere Schranke für $\text{OPT}(\sigma)$.

Wir zeigen $\text{ALG}(\sigma) = \text{OPT}(\sigma)$. Da die von ALG konstruierte Lösung zulässig ist, gilt $\text{ALG}(\sigma) \geq \text{OPT}(\sigma)$. Wir zeigen nun, dass $\text{ALG}(\sigma) \leq \text{OPT}(\sigma)$. Für jedes

Element r_i merken wir uns das oberste Element $w(r_i)$ (in Iteration i) des letzten (am weitesten rechts) Stapels auf den es nicht gelegt werden konnte. Wir nennen $w(r_i)$ den Zeugen von r_i .

Sei x das oberste Element im letzten (weitesten rechts) Stapel nachdem alle Elemente gestapelt wurden. Betrachte die Folge der Zeugen: $(x, w(x), w(w(x)), \dots)$. Die Umkehrfolge definiert eine steigende Teilfolge von σ der Länge $\text{ALG}(\sigma)$ und somit gilt $\text{OPT}(\sigma) \geq \text{ALG}(\sigma)$.

Zu (b): Wir verwenden den Algorithmus aus Aufgabe (a). Stapel die Elemente bzgl. ALG. Sei k die Anzahl der benötigten Stapelplätze (ohne Höhengschränke). Partitioniere jeden Stapelplatz bottom-up in Pakete der Größe h . Sei U die Anzahl der unvollständigen Pakete. Wir haben dann maximal k nicht vollständig gefüllte Pakete d.h., $U \leq k$, und jedes dieser Pakete beinhaltet zumindest ein Element. Sei V die Anzahl der vollständigen Pakete. Es gilt $V \cdot h + U \leq n$. Es ist leicht zu sehen, dass $\text{OPT}(\sigma) \geq \lceil n/h \rceil$. Ferner gilt $\text{OPT}(\sigma) \geq k \geq U$. Somit

$$\text{ALG}(\sigma) \leq U + V \leq U + \frac{n}{h} - \frac{U}{h} = \left(1 - \frac{1}{h}\right)U + \frac{n}{h} \leq \left(2 - \frac{1}{h}\right)\text{OPT}(\sigma).$$

◇