ufgabe 11.2 Online-Interval-Scheduling, leicht

Vir werden nun das bereits behandelte Interval-Scheduling-Problem in seiner Online-Version betrachten. Das heißt wir rhalten eine Eingabe $X = x_1, \ldots, x_n$ mit $x_i = [s_i, f_i[$ und $s_i < s_{i+1}$ und müssen zu jedem x_i entscheiden, ob wir es in unser chedule aufnehmen oder nicht, ohne die Werte für $x_{i+1}, \ldots, x_n$ zu kennen und ohne, dass sich die Jobs in diesem Schedule berschneiden.

1. Wir möchten nun ein möglichst großes Schedule konstruieren. Zeigen Sie, dass es für kein festes $k \in \mathbb{N}$ einen Algorithmus gibt, der k-kompetitiv ist. → Tontan Skript
2. Wir möchte nun nicht mehr ein möglichst großes Schedule produzieren, sondern ein Schedule $S \subseteq \{1, \ldots, n\}$ welches $\sum_{i \in S} |f_i - s_i|$ maximiert, also »möglichst lang« ist. Zeigen Sie, dass es auch für dieses Problem für kein festes $k \in \mathbb{N}$ einen k-kompetitiven Algorithmus gibt.

Aufgabe 11.3 Feuerwehr, mittel

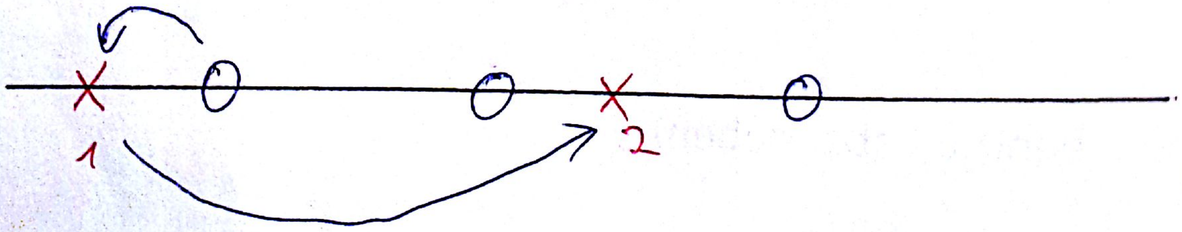
In Linienland, dem einzigen eindimensionalen Staat der Erde, gibt es $k \in \mathbb{N}$ Feuerwehrwagen und eine einzige Straße, die natürlich auch eindimensional ist. Da die Einwohner große Fans des Zirkus sind, ist Feuerspucken in Linienland ein weit verbreitetes Hobby. Daher kommt es im Verlaufe der Zeit an verschiedenen Stellen der Straße zu Bränden. Um dieses zu löschen, muss sich einer der Feuerwehrwagen von seiner aktuellen Position zur Position des Feuers bewegen. Wo es zu Bränden kommt, ist im Vorhinein nicht vorherzusagen. Da Linienland bekanntermaßen wenig natürliche Resourcen besitzt, soll die Brandbekämpfung so organisiert werden, dass die insgesamt zurückgelegte Wegstrecke möglichst klein ist (Die Zeit ist hierbei nicht entscheidend).

1. Formulieren Sie dieses Problem als Optimierungsproblem Π mit Eingaben X und Aktionen Y.
2. Der Premierminister des Landes, Herr Punto, bittet Sie um Hilfe. Entwickeln Sie eine Strategie für die Brandbekämpfung und analysieren Sie die kompetitive Rate Ihrer Strategie.

O Auto

o Brand

Strecke : klein.



Linie

Bewegen, sobald Brand da

1 Brand pro Schritt

<u>Serie 11 UE</u>

$A(x) \le k \cdot Opt(x) + b$ mit $Opt(x) \neq 0$, da sonst k egal.

falls k-kompetitiv:

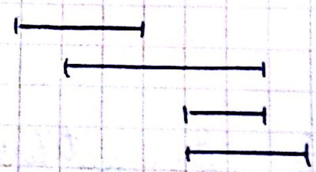
" $\exists$ Algo $\exists k, b$ $\forall \mathcal{X}$: $A(\mathcal{X}) \le k \cdot Opt(\mathcal{X}) + b$ "

Negation:

" $\forall$ Algo $\forall k, b$ $\exists \mathcal{X}$: $A(\mathcal{X}) > k \cdot Opt(\mathcal{X}) + b$ "

↳ über alle Algo (abstrakt), aber fiese Instanz konkret!

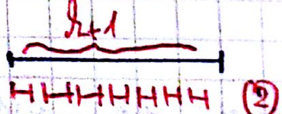
Bsp.:



" Adversary weiß, wie sich $A(\mathcal{X})$ verhält & kann ihn dann platt machen! " A ist deterministisch

Über alle Algo. sehr schwer, aber Instanz bauen.

Job.



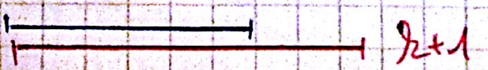
Algo kann: ① nicht nehmen → Adversary: Opt nimmt

$\rightarrow A(\mathcal{X}) = 0, \quad Opt(\mathcal{X}) = 1$

k unbeschränkt!

② Algo nimmt $A(\mathcal{X}) = 1, \quad Opt(\mathcal{X}) = k+1 > k$

→ Optimierungsproblem (Max!) $\frac{Opt(\mathcal{X})}{A(\mathcal{X})} = k+1 > k$



Algo: ① nehmen nicht: $A(\mathcal{X}) = 0$, $Opt(\mathcal{X}) = 1$

② Nehmen: $A(\mathcal{X}) = 1$, Adversary baut Job $k+1$ daneben

$Opt(\mathcal{X}) = k+1$

Maximierungsproblem: $\frac{Opt(\mathcal{X})}{A(\mathcal{X})}$ Minimierung: $\frac{A(\mathcal{X})}{Opt(\mathcal{X})}$

Wenn $A(X) \le T_1$ & $Opt(X) \le T_2$ gilt NICHT $\frac{A(X)}{Opt(X)} \le \frac{T_1}{T_2}$

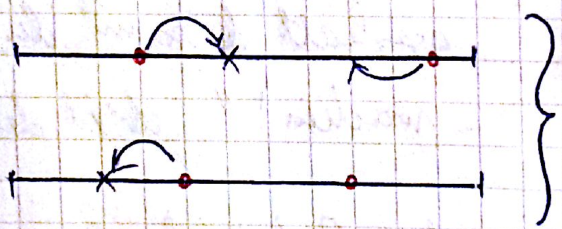
Es gilt auf jeden Fall: $\frac{A(X)}{Opt(X)} \le \frac{T_1}{T_2}$, wenn $A(X) \le T_1$ & $Opt(X) \ge T_2$

Aufg. 3

Komplett korrekt; guter Algorithmus:

DOUBLE COVERAGE

Nächsten hinfahren, falls dann ein Brand zwischen den Autos fahren beide los bis einer da ist.



2- Kompetitiv!

Kompetitive Rate: Zeigen, dass man trotz online-Ungewissheit nur 2-mal schlechter ist als Optimaler, der alles kennt!

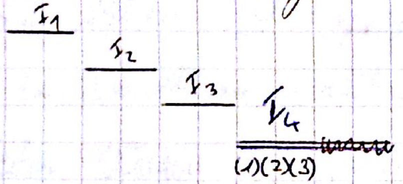
(Greedy-Offline Laufzeit $O(n^2)$, sortieren, p_i, etc.)

Aufg. 2

① Folgender Algorithmus widerlegt, dass eine c-kompetitive Strategie gefunden werden kann für online-Algorithmen:

- Produziere Intervalle $[1,2)$, $[2,3)$, $[3,4)$, ... bis der Algorithmus zum 1. Mal „schedule" entscheidet. ✓
- Anschließend produziere Intervalle der Größe $\frac{1}{c+1}$, die alle nebeneinander innerhalb dieses letzten Intervalls liegen. ✓

→ Wählt der Algorithmus $A(x)$ also z.B. I_4 & die Behauptung lautet, dass bei dieser Wahl der Algorithmus $A(x)$ höchstens z.B. $c=2$ mal schlechter ist,



dann produziert der Adversary nun die Intervalle der Länge $\frac{1}{3}$, die innerhalb von I_4 liegen & die dessen Länge streifen. Statt also $c+1$ Intervalle zu wählen, wählt $A(x)$ also nur 1 Intervall aus & die kompetitive Rate ist schlechter als c. □ ✓

(vgl. Skript Tantau: Algorithmendesign WS 2011)

② Analog verfährt man hier beim Maximierungsproblem der abgedeckten Zeit, um zu zeigen, dass es keinen c-kompetitiven online-Algorithmus zu diesem Problem gibt. Der Adversary produziert dabei immer Intervalle gleichlanger Form mit f_j & s_j & sobald sich $A(x)$ entscheidet einen Job zu schedulen, produziert der Adversary ein Intervall der Länge $(c+1)(f_2 - s_1)$, den $A(x)$ nicht mehr schedulen kann. Erst nach Ablauf dieses Jobs produziert der Adversary wieder seine Standard-Requests.

Die Behauptung der c-Kompetitivität $\frac{A(x)}{Opt(x)} \geq \frac{1}{c}$ bei Maximierungsproblemen gilt also nicht, denn für diesen Adversary gilt: ✓ sehr gut!

$$\frac{A(x)}{Opt(x)} \geq \frac{\sum_{i \in S} |f_i - s_i|}{\sum_{i \in S} (c+1)|f_i - s_i|} \geq \frac{1}{c+1} \quad \square$$

7/7

Aufg. 11.3

(1) Wir gehen davon aus, dass in jedem Schritt ein Brand ausbricht & dementsprechend „Brand-lose Zeiten" einfach entfallen, da auch kein Wagen angefordert wird. ✓

Die Eingaben $X = x_1 x_2 \dots x_n$ mit $x_i \in [0, E]$ geben an, an welcher Stelle im Linienland, das die Grenzen 0 & E hat, ein Brand zum i-ten Schritt ausbricht. ✓

Eine Aktion $Y = y_1 y_2 \dots y_n$ mit $y_i \in \{1, \dots, k\}$ gibt im i-ten Schritt an, welcher der k Wagen den Brand löschen soll. ✓

Jeder Wagen r kennt zum Zeitpunkt i seine Position p_{r_i} im Linienland. Es ist also insgesamt folgender Ausdruck zu minimieren:

$$\sum_{i=1}^{n} \sum_{m=1}^{k} |p_{m_i} - p_{m_{i-1}}| \overset{!}{=} \min$$

d.h. die Abstände aller Fahrzeuge zwischen ihren Positionen zu 2 Zeitschritten

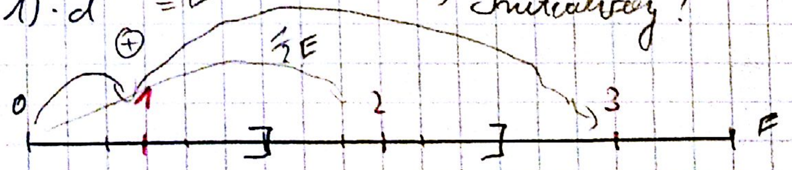
Außerdem gilt, dass zu jedem Schritt eine Feuerposition von einem Wagen übernommen wird $x_i \in \{p_{1_i}, \dots, p_{k_i}\}$

Unser Algorithmus $A(x)$ teilt Linienland in gleichgroße Löschbereiche & platziert die Fahrzeuge im Initialisierungsschritt in der Mitte ihrer Löschbereiche: Bezirksgröße: $d = \frac{E}{k}$,

Anfangsposition des Wagens mit der Nummer $r \in \{1, \dots, k\}$:

$$p_{r_0} = \frac{d}{2} + (r-1) \cdot d \quad = E \quad \Rightarrow \text{Initialisierung!}$$

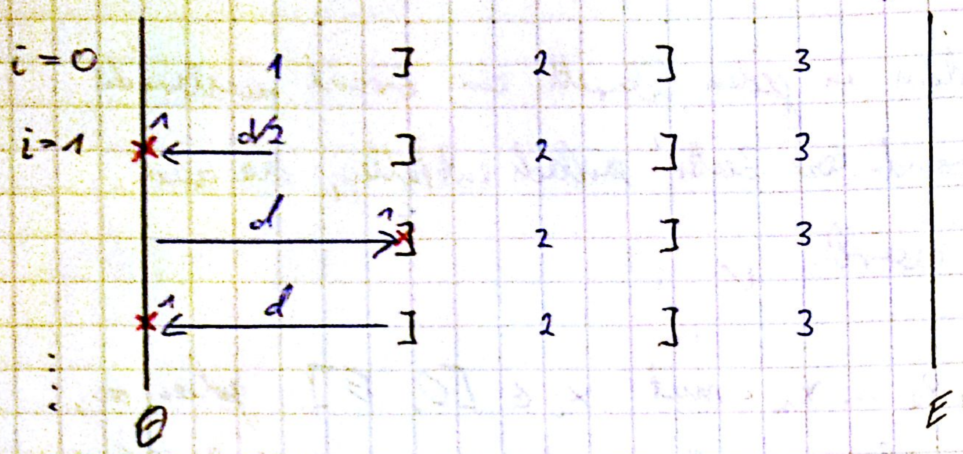
z.B. $k = 3$, $E = 9$



Fällt ein Feuer in den Abschnitt $(x_i \bmod d) + 1$, so kümmert sich der jeweilige Wagen darum ($(5 \bmod 3) + 1 = 2 \rightarrow$ Feuer bei 5: Wagen 2)

O.B.d.A. starten die Wagen von der Zentrale bei $x = 0$ aus, d.h. zur Initialisierung wird $d(\dots) = \frac{k}{2} \cdot E$ zurückgelegt.

Das worst-case-Szenario ist wie folgt:



Also immer Feuer bei O & $\frac{E}{r} = d \rightarrow E = r \cdot d$

Im Fall, dass ein Offline-Algorithmus dies erkennt, muss er nur einen Wagen zur Initialisierung nach d fahren & danach werden ohne Bewegung alle Feuer bei O & d gelöscht

$$CR(Alg(x)) = \frac{\overset{\text{Init}}{\frac{r^2}{2} \cdot d} + d \cdot n}{\underset{\text{Init}}{d} + 0} = \frac{r^2}{2} + n$$

Anzahl der Schritte

$\Longrightarrow$ nicht r kompetitiv (wähle bei n beliebig & $r^2 > r$!)

sehr gut!

7/7

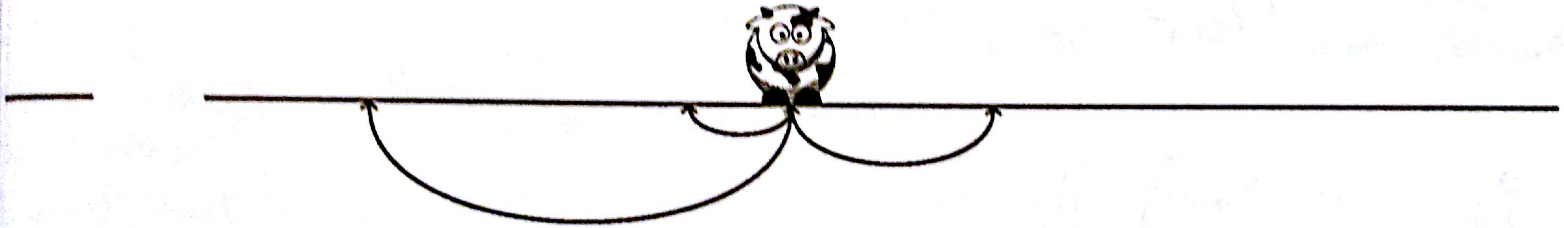
Ziele dieses Blattes

1. Amortisierte Laufzeiten verstehen
2. Methode der Potentialfunktion beherrschen

Hausaufgaben (benotet, abzugeben)

Aufgabe 12.1 Flucht mit Kühen, leicht

Eine Kuh möchte durch einen Zaun hindurch. Sie weiß auch, dass dieser Zaun ein Loch hat. Wo dieses Loch ist, weiß sie nicht. Aufgrund ihrer Ausbildung in Algorithmendesign beschließt die Kuh, zuerst einen Meter nach links zu laufen. Hat sie das Loch nicht gefunden, kehrt sie zum Ausgangspunkt zurück und läuft zwei Meter nach rechts. Hat sie das Loch wiederum nicht gefunden, so kehrt sie zum Ausgangspunkt zurück und läuft vier Meter nach links. Diese Technik der Verdoppelung führt sie solange aus, bis sie das Loch gefunden hat.



1. Was ist der Platz, für den das Verhältnis zwischen optimaler Strecke und zurückgelegter Strecke bei dieser Strategie möglichst schlecht ist?
2. Zeigen Sie, dass die Kuh mit dieser Strategie maximal das 9-fache des optimalen Wegs zurücklegt.
3. Können Sie die Strategie der Kuh verbessern? Nein. → was hinlabern // Nach Wolfram Alpha wird der Ausdruck $1+2m+\sum_{i=0}^{\infty} \frac{1}{m^i}$ bei $m=2$ minimal ⇒ also schon top!

Aufgabe 12.2 Potentialfunktion für Datenstrukturen, mittel

Potentialfunktionen können auch zur Analyse der amortisierten Laufzeit der Operationen einer Datenstruktur verwendet werden. Sei D_i die Datenstruktur nach der i-ten Operation. Sind c_i die Kosten, die D_{i-1} hat, wenn es die i-te Operation ausführt, so definieren wir die *amortisierten* Kosten a_i als

$$a_i = c_i + \Phi(D_i) - \Phi(D_{i-1}).$$

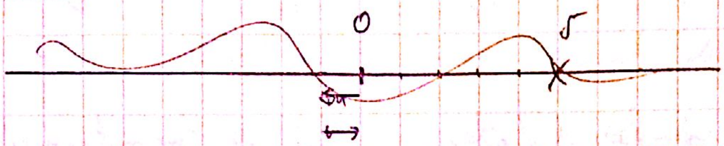
1. Zeigen Sie, dass die Summe der amortisierten Kosten größer oder gleich der Summe der Kosten ist, wenn $\Phi(D_0) = 0$ und $\Phi(D_n) \geq 0$.

... Stacks betrachten. Ein Stack S unterstützt die Operationen PUSH(x), E 3/3 Max Blendowski 592 343

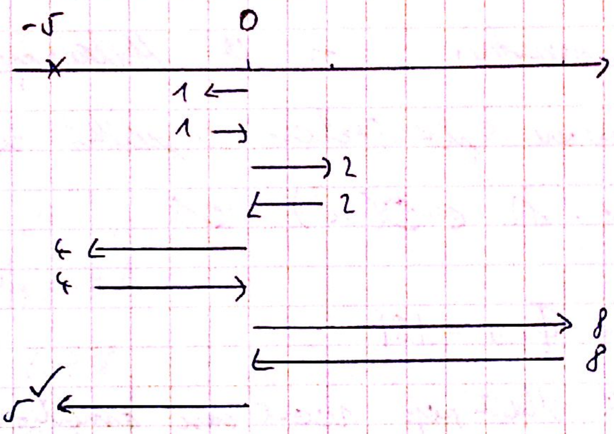
Aufg. 1

① Der Platz ist so gewählt, dass die Kuh noch einmal komplett in die falsche Richtung läuft und der Nenner des Quotienten für die kompetitive Rate möglichst klein wird. ✓

~~Bsp.:~~ ~~n = 5~~



Bsp.: $n = -5$



$\sum_{i=0}^{n+1} 2^i = 2^{n+2} - 1$

im kleiner!

$011\ldots1$
$+1$
$100\ldots0$

② Allg.: Optimal läuft $2^x + 1$ ✓

A läuft: $2^x + 1 + 2 \cdot \sum_{i=0}^{x+1} 2^i$ ✓

$$\Rightarrow CR(A) = \frac{2^x + 1 + 2\cdot\sum_{i=0}^{x+1} 2^i}{2^x + 1} = 1 + \frac{2 \cdot 2^{x+1} + \overbrace{2\cdot\sum_{i=0}^{x} 2^i}^{=\,2^{n+2}-1}}{2^x + 1}$$

$\sum_{i=0}^{x} 2^{i+1}$

$\rightarrow 2^{x+1}$ Einsen!

auch für kleine p!

...eil ausgleicht
...worst-case...
...der wird
...besser!

...s mit
...rechnen!
...ntitieren!

$\Rightarrow$ für große x gilt ~~also~~:

$$CR(A) \cancel{=} \leq 1 + 2\cdot 2 + 2\cdot \sum_{i=0}^{\infty} \left(\tfrac{1}{2}\right)^i = 1 + 4 + 4 = 9 \;✓$$

③ Nach Wolfram Alpha erreicht der Ausdruck $1 + 2m + 2\cdot\sum_{i=0}^{\infty}\left(\frac{1}{m}\right)^i$ für $m = 2$ sein Extremum, also ist der Algorithmus unserer Meinung nach nicht verbesserbar. Gut!

7/7