universität freiburg

Algorithmen und Datenstrukturen SS 2025

Vorlesung 6: Dynamische Felder

Dienstag, 27. Mai 2025

Prof. Dr. Hannah Bast
Professur für Algorithmen und Datenstrukturen
Institut für Informatik
Albert-Ludwigs-Universität Freiburg

Blick über die Vorlesung heute

- Organisatorisches
  - Erfahrungen Ü5 — Hash Maps
  - Zeitmanagement — Lassen wir aus Zeitgründen heute weg
  - Vorgucker Ü6 — Kilometerzähler + Potenzialfunktion
- Inhalt
  - Begriffserklärung — Statische vs. dynamische Felder
  - Realisierung eines dynamischen Feldes — Implementierung + Laufzeitmessung
  - Laufzeitanalyse — Die Laufzeit ist "im Mittel" konstant
  - Alternativer Beweis — Mittels einer Potenzialfunktion

Erfahrungen mit dem Ü5 1/2

- Auszüge aus Ihrem Feedback [halbwegs repräsentativ]

"Echt gute Vorlesung und das Übungsblatt hat Spaß gemacht"

"Vorlesung echt sehr gut strukturiert und verständlich gehalten"

"Sehr cooles Übungsblatt, weil es sehr realitätsnah war"

"Merke, dass mir die Praxis deutlich leichter fällt als die Theorie"

"Sehr coole und spaßige Aufgabe, endlich mal keine Theorie mehr"

"Vorlesung super verständlich, habe trotzdem sehr lange gebraucht"

"Tests waren nicht für offene Adressierung ausgelegt"

"Verwirrt, dass Größe 2000 vorgegeben war und keine Primzahl"

"Ich lese immer das Feedback und freue mich darüber sehr"

"Frustriert, weil gutes Gefühl bei Abgabe, aber dann doch nicht gut"

Erfahrungen mit dem Ü5 2/2

- Was fanden Sie an den Noten besonders interessant

  "Interessant aber nicht überraschend: Durchfallquote in Mathe 1"

  "Wie erwartet bessere Ergebnisse bei fortgeschrittenen Modulen"

  "Dass die Leute entweder in EidP oder Mathe gefaltet werden"

  "Veranstaltungen im 1.+2. Semester mit hohen Durchfallquoten haben einen guten Schnitt … es zeichnen sich zwei Gruppen ab"

  "Bei EidP fallen im SS mehr Leute durch als im WS"

  "In den letzten 5 Jahren scheinbar mehr Leute durchgefallen"

  "Einige Module mit Durchfallquote 80%; kann doch nicht mehr die Schuld der Studierenden sein, dass sie zu wenig gelernt haben"

  "Die Noten bei AlgoDat sehen leider nicht so rosig aus"

  "Bei knapp einem Viertel der Veranstaltungen fällt niemand durch"

Zeitmanagement

- Manchmal fehlt uns am Ende etwas die Zeit

  Warum dann nicht einfach nächste Woche weitermachen?

  Ein Thema pro Vorlesung hat sich sehr bewährt

  Filme und so einen Quatsch am Anfang weglassen

  Fünf Minuten Sinnlosigkeit pro Vorlesung müssen sein

  Bei den letzten Folien geht es manchmal sehr schnell

  1. Versuche ich zu vermeiden, ist aber schwieriger als man denkt, zumal die Vorlesung bewusst sehr interaktiv gestaltet ist

  2. Früher war die Vorlesung perfekt getimed, aber es blieb kaum Zeit zum Nachdenken und es wurden kaum Fehler bemerkt

  3. Das Wichtigste kommt immer im ersten und zweiten Drittel

Vorgucker auf das Ü6

- Implementieren Sie einen "Kilometerzähler"

  A1 Schreiben Sie ein Programm, das von …000 hochzählt

  Nette Aufgabe, braucht man tatsächlich öfter in der Praxis

  A2 Beweisen Sie: Laufzeit für's Hochzählen im Schnitt O(1)

  Mit Hilfe einer geeigneten Potenzialfunktion … was das ist und wie das funktioniert, lernen Sie in der Vorlesung heute

Statische vs. Dynamische Felder 1/3

- Terminologie und Definition
  - **Statisches Feld** "static array" oder "fixed-size array"

    Die Größe des Feldes wird bei der Deklaration festgelegt und kann danach nicht mehr geändert werden

    Unflexibel, dafür sehr einfach und effizient zu realisieren
  - **Dynamische Feld** "dynamic array" oder "variable-size array"

    Die Größe des Feldes kann im Laufe seines Lebens jederzeit beliebig vergrößert oder verkleinert werden

    Flexibel, dafür komplexer und nicht ganz so effizient

    Hinweis: "statisch" kann in der Informatik je nach Kontext verschiedene Sachen bedeuten: "statische Größe" (so benutzen wir es hier), "statische Methode", "statisches Linking", "statisches Typchecking", ...

Statische vs. Dynamische Felder 2/3

- Verwendung in kompilierten Programmiersprachen
  - Da gibt es in praktisch immer beide Typen von Felder
  - Beispiel **Java**

```
int[100] a1;                                // Static, fixed size 100.
var a2 = new ArrayList<Integer>();  // Dynamic, initial size 0.
a2.add(42);                                 // Resizes automatically.
```

  - Beispiel **C++**

```
int[100] a1;                                // Static, fixed size 100.
std::vector a2();                       // Dynamic, initial size 0.
a2.push_back(42);                       // Resizes automatically.
a2.resize(100):                         // Explicitly resize to 100.
```

Statische vs. Dynamische Felder 3/3

- Verwendung in Python (das eine interpretierte Sprache ist)
  - In **Python** gibt es als Teil der Standardsprache nur "list" und das ist ein dyamisches Feld

    Grund: Python hat keine statischen [sic] Typen und ist sowieso nicht effizient und hat in dem Fall Einfachheit bevorzugt
  - In **numpy** (Python Bibliothek für einfache lineare Algebra) gibt es aber effiziente Felder fester Größe, und zwar

    numpy.array

    Das benutzen wir beim Live-Coding gleich

Realisierung Dynamisches Feld 1/5

- Realisierung dynamischer Felder, Grundidee
  - Intern ein fixed-size array (FSA) von ausreichender Größe
  - Wenn Elemente dazu kommen oder entfernt werden, schauen wir, ob die Größe des internen FSA noch passt

    Falls nicht, erzeugen wir ein neues FSA passender Größe und kopieren die Elemente vom alten in das neue FSA

    Diesen Vorgang nennen wir **Reallokation**
  - Das implementieren wir jetzt zusammen

    Wir beschränken uns dabei auf **push** (neues Element ans Ende des Feldes anhängen)

    Analog geht dann auch **pop** (letztes Element entfernen)

Realisierung Dynamisches Feld 2/5

- Vergrößerungsstrategie 1
  - Wir vergrößern das Feld nach jedem push ... und machen es dabei aber immer nur genau **um eins** größer

    Beobachtung: Laufzeit NICHT linear
- Analyse
  - Sei T_i die Laufzeit für das i-te push
  - Dann ist $T_i \geq A \cdot i$ für irgendeine Konstante A

    Bei einer Reallokation müssen alle Elemente kopiert werden

$$\sum_{i=1}^{n} T_i \geq \sum_{i=1}^{n} A \cdot i = A \cdot \sum_{i=1}^{n} i \geq A/2 \cdot n^2$$

$$\sum_{i=1}^{n} i = \frac{1}{2} n(n+1) \geq n^2/2$$

Realisierung Dynamisches Feld 3/5

- Vergrößerungsstrategie 2
  - Wie vorher, aber jetzt bei jedem Vergrößern **um C größer**, für ein festes C, zum Beispiel C = 100 oder C = 1000

    Beobachtung: Immer noch NICHT linear
- Analyse
  - Sei wieder T_i die Laufzeit für das i-te push
  - Dann sind die meisten T_i jetzt O(1)
  - Aber für i = C, 2C, 3C, ... ist nach wie vor $T_i \geq A \cdot i$

$$\sum_{i=1}^{n} T_i \geq \sum_{j=1}^{\lfloor n/C \rfloor} T_{Cj} \geq \sum_{j=1}^{\lfloor n/C \rfloor} A \cdot Cj = A \cdot C \sum_{j=1}^{\lfloor n/C \rfloor} j$$

$$\geq A \cdot C \left(\frac{n}{C}\right)^2 / 2 = \frac{A}{2C} \cdot n^2 \qquad n/C \in \mathbb{N}$$

$$\lfloor n/C \rfloor \cdot (\lfloor n/C \rfloor + 1)/2 \geq \lfloor n/C \rfloor^2 / 2$$

Realisierung Dynamisches Feld 4/5

- Vergrößerungsstrategie 3
  - Wie vorher, aber jetzt machen wir bei jedem Vergrößern das Feld genau **doppelt so groß** wie vorher

    Beobachtung: JETZT ist es linear

- Analyse
  - Jetzt Reallokationen nur noch bei $i = 1, 2, 4, 8, 16, \ldots$

    Bei den Reallokationen $T_i \leq A \cdot i$, sonst $T_i \leq A$

$$\sum_{i=1}^{n} T_i \leq A \cdot n + \sum_{j=0}^{\lfloor \log_2 n \rfloor} T_{2^j} \leq A \cdot n + \sum_{j=0}^{\lfloor \log_2 n \rfloor} A \cdot 2^j$$

$$= A \cdot \left(2^{\lfloor \log_2 n \rfloor + 1} - 1\right) \leq 2^{\log_2 n + 1} = 2n$$

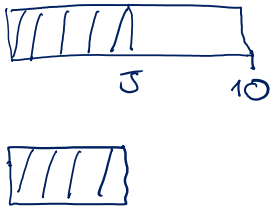
$$\leq A \cdot n + A \cdot 2n = 3A \cdot n$$

Realisierung Dynamisches Feld 5/5

- Entfernen von Elementen
  - Analog zum Vergrößern, könnten wir das Feld auf die Hälfte verkleinern wenn es nur noch halbvoll ist

    Aber: wenn man danach zwei mal **push** macht, muss man das Feld gleich wieder vergrößern

    Und: wenn man danach zwei mal **pop** macht, muss man das Feld gleich wieder verkleinern



  - Deswegen machen wir es so … Details siehe Folie 17

    Wenn Feld **ganz voll** → Größe **verdoppeln**

    Wenn Feld **viertel voll** → Größe **halbieren**

Laufzeitanalyse 1/7

- Vorbetrachtungen
  - Unsere bisherigen Laufzeitanalysen haben angenommen, dass wir nur Elemente hinzufügen (mit push)
  - Dann können wir leicht ausrechnen, wann es zu einer Reallokation kommt

    Mit der Verdoppelungstrategie bei: 1, 2, 4, 8, …
  - Im Folgenden wollen wir unsere Analyse verallgemeinern auf beliebige Abfolgen von push und pop

    Dann können wir nicht mehr so leicht vorhersagen, wann realloziert werden muss

Laufzeitanalyse 2/7

s = size
c = capacity

- Notation
  - Gegeben n Operationen $O_1, \ldots, O_n$

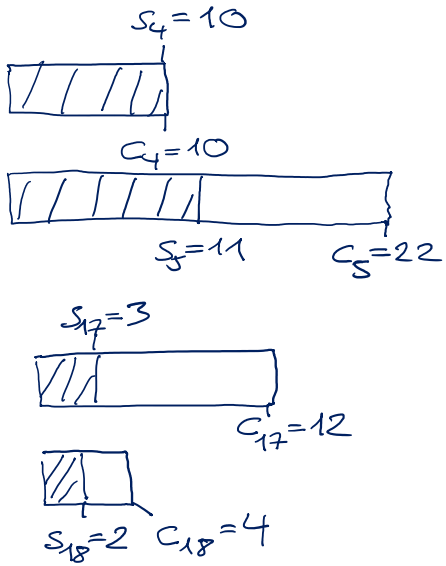
    Eine beliebige Abfolge von **push** und **pop**
  - Sei s_i die Anzahl Elemente **nach** Operation O_i ($s_0 := 0$) $s_1 = 1 \ldots$
  - Sei c_i die Größe des Feldes **nach** Operation O_i ($c_0 := 0$) $c_1 = 2 \ldots$

    Es gilt immer $c_i \geq s_i$ (Feld muss immer "groß genug" sein)
  - Sei wieder T_i die Zeit für Operation O_i
    - Falls Reallokation nicht nötig: $T_i \leq A$
    - Falls Reallokation nötig: $T_i \leq A + B \cdot s_i$

    für irgendwelche Konstanten A und B (unabhängig von n)

Laufzeitanalyse 3/7

- Wir analysieren folgende Variante
  - Falls Operation O_i ein **push** ist:
    - Reallokation genau dann, wenn $s_{i-1} = c_{i-1}$
    - Vergrößerung so, dass danach $c_i = 2 \cdot s_i$
  - Falls Operation O_i ein **pop** ist:
    - Reallokation genau dann, wenn $4 \cdot s_{i-1} \leq c_{i-1}$
    - Verkleinerung so, dass danach $c_i = 2 \cdot s_i$



  - In beiden Fällen ist also **direkt nach** der Reallokation das interne Feld **exakt** doppelt so groß ($c_i = 2 \cdot s_i$)

    Das muss nicht so ein, wird uns aber die Analyse erleichtern
  - Das Feld ist immer zu mindestens **einem Viertel** voll

Laufzeitanalyse 4/7

- Beweisidee
  - **Beobachtung:** teuer sind nur die Operationen, bei denen realloziert werden muss und vor so einer teuren Operation gibt es genügend billige Operationen (ohne Reallokation)
  - **Idee:** Stellen wir uns vor, die Operationen würden Geld kosten und wir hätten einen festen Etat von **100 €** pro Operation

    Nehmen wir an, die billigen Operationen kosten **40 €**

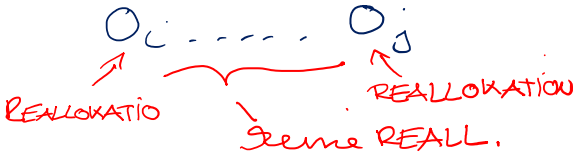
    Dann können wir bei jeder billigen Operation **60 €** auf ein "Sparkonto" legen

    Wenn dann eine teure Operation kommt, nehmen wir das Geld vom Sparkonto (und hoffen, dass es reicht)

    Wenn das klappt, kosten **n** Operation höchsten **n · 100 €**
    (obwohl einzelne Operationen viel teurer als 100 € sein können)

Laufzeitanalyse 5/7

- Wir beweisen zuerst folgenden Hilfssatz



  - Wenn bei O_i eine Reallokation stattfindet, und die nächste Reallokation danach bei O_j, dann ist $j - i \geq s_j/2$

    In der Mathematik nennt man das "Lemma": ein Hilfssatz für das, was wir eigentlich beweisen wollen (siehe Folie 21)

  - Was dieses Lemma intuitiv (mit Hinblick auf unsere Beweisidee von der vorherigen Folie) bedeutet:

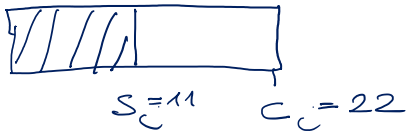
    Wenn wir bei jeder der $j - i$ Operationen "dazwischen" einen konstanten Wert B' ansparen (wir nehmen an, dass wir auch bei den teuren Operationen ansparen), haben wir bei O_j angespart: $B' \cdot (j - i) \geq B'/2 \cdot s_j$

    Die Kosten von O_j sind $\leq A + B \cdot s_j$, bei genügend großem B' würde das Angesparte also reichen

Laufzeitanalyse 6/7

- Beweis des Lemmas
  - Zu zeigen: Wenn bei O_i eine Reallokation stattfindet, und die nächste Reallokation danach bei O_j, dann $j - i \geq s_j / 2$
  - Nach O_i ist auf jeden Fall $c_i = 2 \cdot s_i$

    Egal ob es ein **push** oder ein **pop** war



  - Nächste Vergrößerung nach frühestens $s_i + 1$ Operationen und dann ist $s_j = 2 \cdot s_i + 1$ $\quad s_i = (s_j - 1)/2 = s_j/2 - 0.5$

    In dem Fall ist also $j - i \geq s_i + 1 \geq s_j / 2$ $\quad s_j/2 - 0.5 + 1$
  - Nächste Verkleinerung nach frühestens $s_i/2$ Operationen und dann ist $s_j \leq s_i/2$

    In dem Fall ist also $j - i \geq s_i/2 \geq s_j \geq s_j/2$
  - Damit ist in jedem Fall $j - i \geq s_j / 2$

Laufzeitanalyse 7/7

- Beweis der eigentlichen Aussage

  TELESKOPSUMME:
  $(i_2 - i_1) + (i_3 - i_2) + (i_4 - i_3) + (i_5 - i_4) = i_5 - i_1$

  – **Satz:** Seien die Kosten einer Operation O_i ohne Reallokation $T_i \le A$ und mit Reallokation $T_i \le A + B \cdot s_i$, dann ist

  $i_2 - i_1$, $i_3 - i_2$, $i_4 - i_3$, $i_5 - i_4$

  i_1 i_2 i_3 i_4 i_5

  $$\mathbf{T_1 + T_2 + \ldots + T_n \le (A + 3B) \cdot n}$$

  Eine Operation kostet also im Schnitt nur $\le A + 3B = O(1)$

  – **Beweis:** Seien O_{i_1}, O_{i_2}, …, O_{i_k} die Operationen, bei denen realloziert wird (in chronologischer Reihenfolge, also $i_1 < i_2 < \ldots < i_k$)

  O_7 O_{23} O_{121}

  Dann $T_1 + T_2 + \ldots + T_n \le A \cdot n + B \cdot (s_{i_1} + s_{i_2} + \ldots + s_{i_k})$

  $\Rightarrow s_{i_2} \le 2(i_2 - i_1)$ $s_{i_3} \le 2(i_3 - i_2)$

  Laut Lemma $i_2 - i_1 \ge s_{i_2}/2$ und $i_3 - i_2 \ge s_{i_3}/2$ und …

  Damit $s_{i_1} + s_{i_2} + \ldots + s_{i_k} \le s_{i_1} + 2 \cdot (i_2 - i_1) + 2 \cdot (i_3 - i_2) + \ldots + 2 \cdot (i_k - i_{k-1})$

  $$\sum_{i=1}^{n} T_i \le A \cdot n + B \cdot (s_{i_1} + 2 i_2 - 2 i_1) \le A \cdot n + 3B \cdot n$$

  $s_{i_1} \le n$, $2 i_2 \le n$

Beweis mit Potenzialfunktion 1/5

- Variante des Beweises
  - Der Beweis auf den vorherigen Folien hat die Kosten für eine Folge von Operationen quasi "zu Fuß" analysiert
  - Man kann solche Beweise auch mit Hilfe einer sogenannten **Potenzialfunktion** führen

    Der Wert der Potenzialfunktion entspricht dabei intuitiv dem aktuellen "Kontostand"

    Insbesondere darf eine Operation nur dann teuer sein, wenn entsprechend viel auf dem Konto ist

    Aufgabe 2 vom Ü6 ist eine sehr schöne Übung dafür

Beweis mit Potenzialfunktion 2/5

- Potenzialfunktionen, Mastertheorem
  - Gegeben eine Folge von n Operationen $O_1, \ldots, O_n$ auf einer beliebigen Datenstruktur
  - Sei Φ eine Potenzialfunktion, wobei Φ_i = Wert der **nach** O_i und Φ_0 = Wert am Anfang und alle $\Phi_i \geq 0$
  - Sei T_i die Laufzeit für O_i mit $T_i \leq A + B \cdot (\Phi_{i-1} - \Phi_i)$

    1750€ 250€

    Intuition: wenn $\Phi_{i-1} - \Phi_i$ groß ist, darf auch T_i groß sein
  - Dann ist die Gesamtlaufzeit $\Sigma\, T_i = O(n + \Phi_0)$
  - **Beweis:**

$$\sum_{i=1}^{n} T_i \leq A \cdot n + B \cdot \sum_{i=1}^{n} \Phi_{i-1} - \Phi_i$$

$$= (\Phi_0 - \Phi_1) + (\Phi_1 - \Phi_2) + \cdots$$

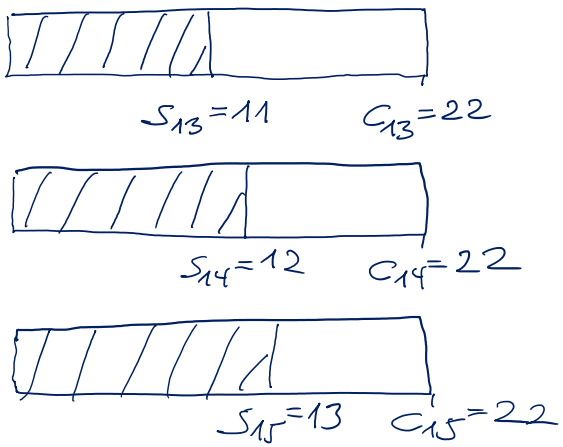
$$\leq A \cdot n + B \cdot (\Phi_0 - \Phi_n)$$

$$\geq 0$$

Beweis mit Potenzialfunktion 3/5

- Anwendung des Satzes für dynamische Felder
  - Wie vorher s_i = Anz. Elemente und c_i = Kapazität **nach** O_i
  - Wir definieren unsere Potenzialfunktion so, dass sie direkt nach einer Reallokation 0 ist und danach langsam wächst:

    $\mathbf{\Phi_i := |\, s_i - c_i / 2 \,|}$
  - Wir zeigen auf der nächsten Folie: $T_i \leq A' + B' \cdot (\Phi_{i-1} - \Phi_i)$
  - Gemäß Mastertheorem ist dann $\Sigma\, T_i = O(n + \Phi_n) = O(n)$



Beweis mit Potenzialfunktion 4/5

- Beweis, dass $T_i \leq A' + B' \cdot (\Phi_{i-1} - \Phi_i)$
  - Wir unterscheiden zwei Fälle:

    Fall 1: O_i ist "billig" (ohne Reallokation) ... $T_i \leq A$

    Potenzial ändert sich um höchstens 1

    Damit $\Phi_{i-1} - \Phi_i \geq -1$ und also $T_i \leq 2A - A \leq 2A + A \cdot (\Phi_{i-1} - \Phi_i)$

    Fall 2: O_i ist "teuer" (mit Reallokation) ... $T_i \leq A + B \cdot s_i$

    Potenzial vor Reallokation: $\geq s_j/2$

    wie beim „zu Fuß" Beweis

    Potenzial nach Reallokation: 0

    Damit $\Phi_{i-1} - \Phi_i \geq s_j/2$

    Also $T_i \leq A + B \cdot s_j \leq A + 2B \cdot (\Phi_{i-1} - \Phi_i)$ ☐

Beweis mit Potenzialfunktion 5/5

- Vergleich der beiden Beweise
  - Beide Beweise waren ungefähr gleich schwierig, aber der Beweis mit Potenzialfunktion hat zwei Vorteile

    1. Der Potenzialfunktion kann man eine intuitive Bedeutung zuschreiben (hier: wie "unentspannt" unserer Feld gerade ist)

    2. Wir können am Ende das Mastertheorem anwenden und wissen dann sofort, dass $\Sigma_i T_i = O(n + \Phi_0)$
  - Es gibt in der Regel viele verschiedene Potenzialfunktionen, für die der Beweis funktioniert (schon wegen der Konstanten)

    Oft ist aber eine davon besonders elegant bzw. natürlich

    Beachten Sie die ausführlichen Hinweise zum Ü6, Aufgabe 2

Literatur / Links

- Dynamische Felder: Laufzeitanalyse
  - In Mehlhorn/Sanders:

    3.2 Unbounded Arrays
  - In Wikipedia

    http://en.wikipedia.org/wiki/Dynamic_array
  - Potenzial vs. Potential

    http://www.duden.de/rechtschreibung/potenzial