Mehrheitsbestimmung
Wer wird Klassensprecher?

Thomas Erlebach
University of Leicester

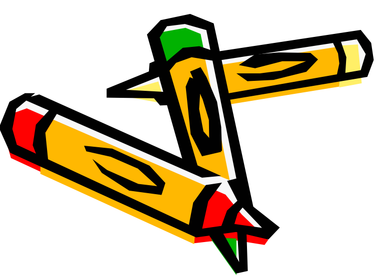
INFORMATIK JAHR
Wissenschaftsjahr 2006

Szenario: Klassensprecherwahl

- Alle schreiben den Namen des von ihnen favorisierten Kandidaten auf einen Zettel.
- Vereinbarung: Der Klassensprecher bzw. die Klassensprecherin gilt nur dann als gewählt, wenn **mehr als die Hälfte** aller Schüler in der Klasse für ihn bzw. sie gestimmt haben.

Aufgabe:
Auszählen der Stimmen

- Um das Wahlergebnis zu bestimmen, muss man also herausfinden, ob irgendjemand mehr als die Hälfte aller Stimmen bekommen hat.
- Wie soll man vorgehen, um diese Aufgabe mit möglichst wenig Aufwand zu lösen?



Erster Ansatz: Stimmen aller Kandidaten zählen

- Führe eine Liste (anfangs leer) und bearbeite die Wahlzettel einen nach dem anderen.
- Wenn auf dem aktuellen Zettel der Name X steht, so gehe wie folgt vor:
  - Falls X noch nicht in der Liste ist, nimm X in die Liste auf und mache einen Strich bei X.
  - Andernfalls mache einfach einen zusätzlichen Strich bei X.

Erster Ansatz: Stimmen aller Kandidaten zählen (2)

- Am Ende sehen wir nach, welcher Name in der Liste die meisten Striche hat.
- Sind dies bei N Schülern mehr als N/2 Striche, so hat dieser Name die absolute Mehrheit der Stimmen erhalten.
- Andernfalls hat niemand die absolute Mehrheit erzielt.

Mögliches Ergebnis: Monika ist Klassensprecherin

Tobias	~~IIII~~
Anton	II
Heinz	I
Corinna	II
Kevin	II
Monika	~~IIII~~ ~~IIII~~ IIII
Laura	I

Kritik

- Der Anatz funktioniert, wäre aber bei einer Wahl mit sehr vielen Kandidaten recht aufwendig:
  - Bei jedem Wahlzettel muss man in der Liste nachsehen, ob der Name schon darin vorkommt.
  - Wenn sehr viele Namen in der Liste sind (und diese vielleicht auch noch schwierig zu vergleichen sind, z.B. Namen mit chinesischen Schriftzeichen), kann das schon mal sehr lange dauern!

Geht es effizienter?

Anderer Ansatz

- Verwende einen Stapel, anfangs leer.
- Phase 1: Bei jedem Wahlzettel:
  - Falls der Stapel leer ist oder wenn der Name X auf dem Wahlzettel gleich dem Namen oben auf dem Stapel ist, so lege den Wahlzettel auf den Stapel.
  - Andernfalls nimm einen Wahlzettel vom Stapel herunter.

Anderer Ansatz (Fortsetzung)

- Falls der Stapel am Ende von Phase 1 leer ist, gib aus: Niemand hat die Mehrheit.
- Andernfalls führe Phase 2 aus:
  - T := Name auf dem Wahlzettel, der oben am Stapel liegt
  - Zähle, wie oft T unter allen Wahlzetteln vorkommt
  - Kommt T mehr als N/2 mal vor, so hat T die absolute Mehrheit; andernfalls hat niemand die absolute Mehrheit erreicht.

Beispiel:
B,B,A,A,C,A,A

Stapel

Beispiel:

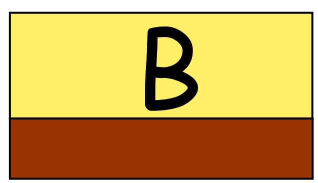
$\boxed{B}$, B, A, A, C, A, A

B

Stapel

Beispiel:

$\boxed{B}$,B,A,A,C,A,A



Stapel

Beispiel:

B,B,A,A,C,A,A

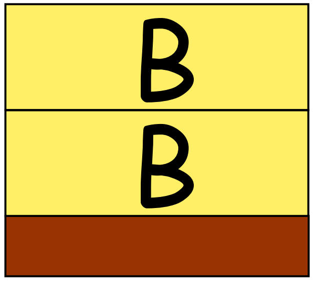
B

B

Stapel

Beispiel:

B,$\boxed{B}$,A,A,C,A,A



Stapel

Beispiel:

B, B, $\boxed{A}$, A, C, A, A

A

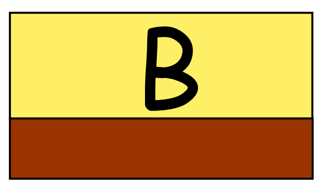
B

B

Stapel

Beispiel:

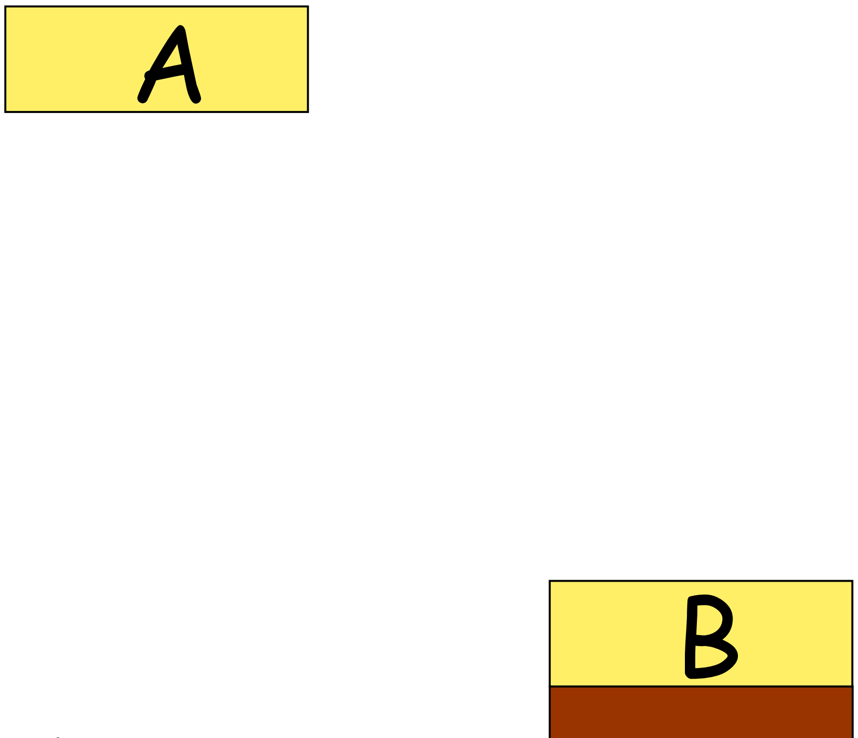
B, B, A, A, C, A, A



Stapel

Beispiel:

B,B,A,$\boxed{A}$,C,A,A



Stapel

Beispiel:

B,B,A,$\boxed{A}$,C,A,A

Stapel

Beispiel:

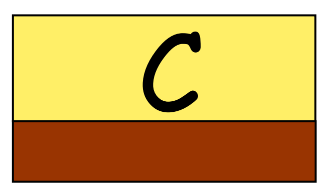
B,B,A,A,$\boxed{C}$,A,A

C

Stapel

Beispiel:

B,B,A,A,C,A,A



Stapel

Beispiel:

B,B,A,A,C,$\boxed{A}$,A

A

C

Stapel

Beispiel:

B,B,A,A,C,A,A

Stapel

Beispiel:

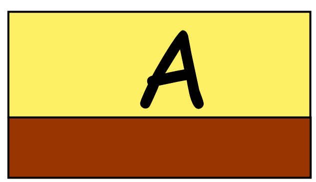
B,B,A,A,C,A,$\boxed{\text{A}}$

A

Stapel

Beispiel:

B,B,A,A,C,A,$\boxed{A}$

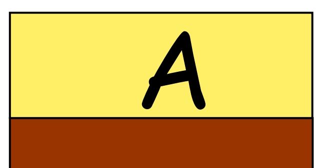


Stapel

Beispiel:
B,B,A,A,C,A,A

Am Ende von Phase 1 liegt ein A oben auf dem Stapel.

In Phase 2 zählt der Algorithmus, wie oft A insgesamt vorkommt, und erkennt A somit richtig als Mehrheitselement.



Stapel

Wieso funktioniert dieser Algorithmus?

- Wir beobachten: Der Algorithmus könnte nur dann einen Fehler machen, wenn die Eingabe ein Mehrheitselement X enthält, am Ende von Phase 1 aber der Stapel leer ist oder ein anderes Element als X oben am Stapel liegt.

Kann dieser Fall jemals eintreten?

Nehmen wir einmal an, dass der Fall tatsächlich eintritt.

Dann liegt nach Phase 1 kein X oben am Stapel

- Was ist mit jedem X in Phase 1 passiert?
  1. Als X bearbeitet wurde, lag gerade ein anderes Element Y oben am Stapel. X wurde nicht auf den Stapel gelegt und Y vom Stapel entfernt.
  2. Als X bearbeitet wurde, wurde es auf den Stapel gelegt, aber später aufgrund eines anderen Elementes Z wieder vom Stapel entfernt.

Abschluss des Beweises

- Jedem X lässt sich somit eindeutig ein Element Y oder Z zuordnen.
- Es gibt daher mindestens so viele Elemente ungleich X, wie es Elemente X gibt.
- Das kann nicht sein, wenn X wirklich das Mehrheitselement ist.

Der Algorithmus ist also immer korrekt!

Ist Phase 2 überhaupt nötig?

- Ja, denn z.B. bei der Eingabe
  A,B,A,C,C
  liegt nach Phase 1 ein C oben auf dem Stapel, und erst durch das „Nachzählen" in Phase 2 merkt der Algorithmus, dass C kein Mehrheitselement ist.

Wie viele Vergleiche macht der Algorithmus?

- Bei einer Eingabe mit n Elementen macht der Algorithmus
  - In Phase 1: höchstens n-1 Vergleiche, da das erste Element ohne Vergleich auf den Stapel gelegt wird und jedes andere Element höchstens mit dem Element oben am Stapel verglichen wird.
  - In Phase 2: ebenfalls höchstens n-1 Vergleiche.
  - Insgesamt **höchstens 2n-2 Vergleiche.**

Geht es besser?

- Ja: Eine verfeinerte Version des Algorithmus, die mit 2 Stapeln arbeitet, benötigt höchstens $\lceil 3N/2 \rceil - 2$ Vergleiche.
- Geht es noch besser? Nein!
  - Man kann zeigen, dass **jeder Algorithmus** auf manchen Eingaben mindestens $\lceil 3N/2 \rceil - 2$ Vergleiche machen muss!

Andere Anwendungen der Mehrheitsbestimmung

- Wichtige Berechnungen von n Prozessoren ausführen lassen und dann das Mehrheitsergebnis verwenden.
- Beobachtung des Datenverkehrs im Internet: Identifizierung **häufiger Elemente** (z.B. Benutzer, die einen Großteil des Verkehrs verursachen).
  - Hier kann eine verallgemeinerte Version des Algorithmus zur Mehrheitsbestimmung eingesetzt werden.

Quelle

- Der Algorithmus zur Mehrheitsbestimmung mit $\lceil 3N/2 \rceil - 2$ Vergleichen (und auch der Beweis, dass es nicht besser geht) wurde bereits 1982 von **M.J. Fischer und S.L. Salzberg** im *Journal of Algorithms* vorgestellt.