

5. Eine untere Schranke für die Medianbestimmung

Satz 88

Jeder (vergleichsbasierte) Medialgorithmus benötigt im worst-case mindestens $\lceil \frac{3n}{2} \rceil - 2$ Vergleiche.

Beweis:

Gegenspielerargument (adversary argument)

n Elemente, o.B.d.A. n ungerade, alle Elemente paarweise verschieden. Die Menge aller Elemente wird in drei Teilmengen partitioniert: U enthält die Kandidaten für den Median, G enthält Elemente, die sicher größer als der Median sind, und L enthält Elemente, die sicher kleiner als der Median sind. Anfangs sind alle Elemente in U .

Beweis (Forts.):

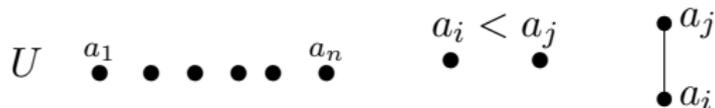
Der Algorithmus stellt nun Fragen der Form $a_i < a_j$. Der Gegenspieler gibt konsistente Antworten, die den Algorithmus jedoch dazu zwingen, möglichst viele Fragen stellen zu müssen, bevor die Antwort feststehen kann (d.h. U soll möglichst ungeordnet bleiben).

Durch die (konsistenten!) Antworten des Gegenspielers auf die " $<^?$ "-Queries des Algorithmus entsteht ein DAG. Der Gegenspieler hält diesen DAG "einfach", z.B. angenommen $y > z$ und $y > x$, dann soll y „sehr groß“ sein $\Rightarrow y \rightarrow G$.

Strategie des Gegenspielers

Beweis (Forts.):

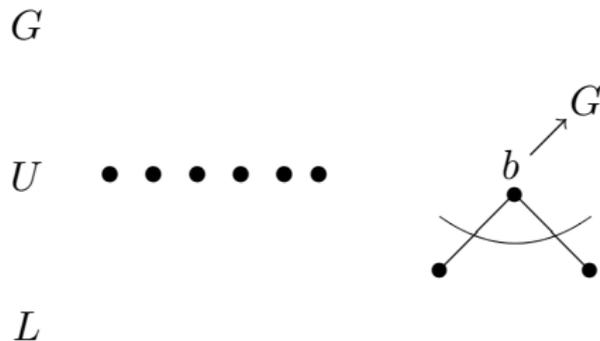
G



L

Strategie des Gegenspielers

Beweis (Forts.):



Strategie des Gegenspielers

Beweis (Forts.):

$$G \leq \frac{n+1}{2} - 1$$

$U \quad \bullet \bullet \bullet \bullet \bullet$



Solange ein Element unverglichen ist, ist nicht klar, welches der Median ist.

$$L \leq \frac{n+1}{2} - 1$$

Beweis (Forts.):

Solange $|L|, |G| < \frac{n+1}{2}$, kann der Algorithmus annehmen, dass der Median in U ist. Solange U mindestens zwei unverglichene Elemente **und** keine Zusammenhangskomponente mit > 2 Elementen enthält, kann der Algorithmus den Median **nicht** bestimmen.

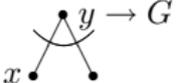
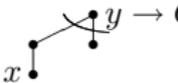
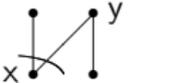
Strategie des Gegenspielers

Beweis (Forts.):

Die Strategie des Gegenspielers hat zwei Phasen.

Erste Phase: Query sei $x \stackrel{?}{<} y$:

- i) $x, y \in G$ bzw. $x, y \in L$: irgendeine konsistente Antwort
- ii) $x \in G \wedge y \in L \cup U$ (bzw. $x \in L \wedge y \in G \cup U$):
Antwort: $y < x$ (bzw. $x < y$).
- iii) Sei $x, y \in U$.

	Query	Antwort des Gegenspielers	Anzahl der Paare in U	$ U $	$ L $	$ G $
1.	$x \dots y$		+1	—	—	—
2.	$x \dots y$		-1	-1	0	+1
3.	$x \dots y$		-1	-1	1	0
4.	$x \dots y$		-1	-1	0	+1
5.	$x \dots y$		-1	-1	+1	0
6.	$x \dots y$		-1	-1	+1	0

Die erste Phase endet, wenn $|L| = \frac{n-1}{2}$ oder $|G| = \frac{n-1}{2}$. Während der Phase 1 enthält U mindestens zwei (in U) maximale und mindestens zwei (in U) minimale Elemente (bzgl. des DAGs). \Rightarrow Während Phase 1 kann der Algorithmus den Median mit Sicherheit **nicht** bestimmen.

Der Gegenspieler beginnt mit Phase 2, sobald

$$|L| \text{ wird } \frac{n-1}{2} \text{ oder } |G| \text{ wird } \frac{n-1}{2}.$$

O.B.d.A.:

$$|L| = \frac{n-1}{2}$$

Der Gegenspieler zwingt nun den Algorithmus, das minimale Element in U bzgl. der gesamten totalen Ordnung zu bestimmen (da dieses unter den Vorgaben der Median ist).

Beweis (Forts.):

Laufzeitanalyse:

C := Anzahl der Vergleiche

P := Anzahl der Paare in U

- i) In der Phase 1 gilt folgende Invariante: $C - P + 2|U| \geq 2n$.
Dies wird durch vollständige Induktion gezeigt:
Induktionsanfang: $0 - 0 + 2n \geq 2n$;
Induktionsschritt: Gemäß der Tabelle.
- ii) Sei C die Anzahl der Vergleiche am Ende der Phase 1. In der Phase 2 werden noch $\geq |U| - 1 - |P|$ Vergleiche nötig. Die Anzahl der Vergleiche für alle Phasen ist damit

$$\geq C + |U| - 1 - |P|.$$

Beweis (Forts.):

Am Ende der Phase 1 gilt ja $C \geq 2n + |P| - 2|U|$ (wg. Invariante).
Damit gilt für die Anzahl der Vergleiche:

$$\begin{aligned} &\geq 2n + |P| - 2|U| + |U| - 1 - |P| = 2n - |U| - 1 \\ &\geq \frac{3}{2}n - \frac{3}{2} = \\ &= \left\lceil \frac{3}{2}n - 2 \right\rceil, \text{ da } |U| \leq \frac{n+1}{2} \end{aligned}$$

□

6. Eine bessere untere Schranke

Satz 89

Sei T ein Entscheidungsbaum für die Bestimmung des i -kleinsten von n verschiedenen Elements, mit $i^2 \geq \log \left[\binom{n}{i} \frac{1}{n-i+1} \right] + 3$. Dann gilt, wenn

$$p := 2\sqrt{\log \left[\binom{n}{i} \frac{1}{n-i+1} \right] + 3} - 2$$

gesetzt wird,

$$\text{Höhe}(T) \geq \log \left[\binom{n}{i} \frac{2^{n-p}}{n-i+1} \right].$$

Bemerkung: $i = \lceil \frac{n}{2} \rceil \rightarrow \binom{n}{\lceil \frac{n}{2} \rceil} = \Theta\left(\frac{2^n}{\sqrt{n}}\right)$; also erhalten wir eine untere Schranke von $\text{Höhe}(T) \geq 2n - o(n)$ für die Bestimmung des Medians.

Beweis:

Der Beweis setzt sich aus einem Gegenspieler- und einem Abzählargument zusammen, um zu zeigen

„ T hat viele Blätter.“

Sei A Teilmenge der Schlüssel, $|A| = i$. Wir konstruieren einen Teilbaum T_A von T , so dass alle Blätter von T_A auch Blätter von T sind, und zeigen: T_A hat viele Blätter (nämlich 2^{n-p}). Es gibt $\binom{n}{i}$ Möglichkeiten, A zu wählen, jedes Blatt von T kommt in höchstens $n - i + 1$ T_A 's vor.

Beweis (Forts.):

Beobachtungen:

Jedes Blatt w von T liefert folgende Informationen:

$\text{answer}(w) \xrightarrow{\text{liefert}} i\text{-kleinstes Element } x$

$\text{little}(w) \xrightarrow{\text{liefert}} i - 1 \text{ Elemente } < x$

$\text{big}(w) \xrightarrow{\text{liefert}} n - i \text{ Elemente } > x$

Beweis (Forts.):

Wähle A als beliebige Teilmenge der n gegebenen Schlüssel, mit $|A| = i$. Wir geben für den Gegenspieler eine Strategie an, welche dazu führt, dass wir durch Zurechtschneiden aus T einen Baum T_A konstruieren können, so dass gilt:

- T_A ist Binärbaum
- jedes Blatt von T_A ist auch Blatt von T , d.h. durch das Zurechtschneiden entstehen keine zusätzlichen Blätter
- für jedes Blatt w von T_A gilt: $\text{little}(w) \subset A$

Beweis (Forts.):

Setze \bar{A} gleich dem Komplement von A , sowie

$$r := \sqrt{\log \left[\binom{n}{i} \frac{1}{n-i+1} \right]} + 3$$
$$s := r - 1$$

Damit gilt: $p = r + s - 1$.

Die Konstruktion (das “Zurechtstutzen“) von T zu T_A erfolgt in zwei Phasen.

Beweis (Forts.):

Erste Phase: Breitensuche von der Wurzel nach unten.

Betrachte Knoten x . Definiere:

- $C(x)$ sind die Elemente $a \in A$, für die es kein $b \in A$ und keinen Knoten auf dem Pfad von der Wurzel von T zu x gibt, an dem a mit b mit dem Ergebnis $a < b$ verglichen wurde.
- $c(x)$ sind entsprechend die im Knoten x bekannten Minima in \bar{A} .
- $s(x, a)$ ist, für $a \in A$, die Anzahl der Elemente $\in c(x) \subseteq \bar{A}$, mit denen a auf dem Pfad von der Wurzel zu x verglichen wurde.

