Beweis (Forts.):

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

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

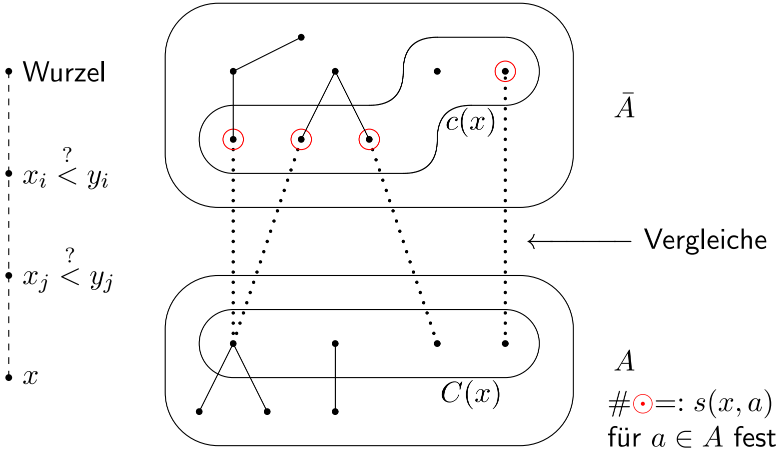
Damit gilt: $p = r + s - 1$.

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.


Wurzel
$x_i \overset{?}{<} y_i$
$x_j \overset{?}{<} y_j$
x
$c(x)$
$\bar{A}$
Vergleiche
A
$C(x)$
$\#\odot =: s(x,a)$
für $a \in A$ fest

Beweis (Forts.):

Regeln für Phase 1:

Seien a und b die Elemente, die im Knoten x verglichen werden.

1.1: Falls $a \in A$ und $b \in A$, behalte x in T_A bei.

1.2: Falls $a \in \bar{A}$ und $b \in \bar{A}$, behalte x in T_A bei.

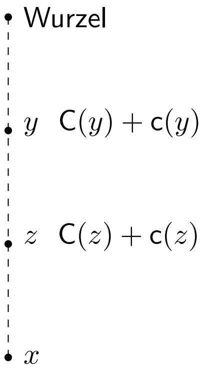
1.3: Sei nun o.B.d.A. $a \in A$ und $b \in \bar{A}$. Ersetze den Unterbaum in T mit Wurzel x mit dem Unterbaum, dessen Wurzel das Kind von x ist, das dem Ergebnis „$a < b$“ entspricht (d.h. lasse den Vergleich $a < b$ aus, da gemäß Strategie alle Elemente aus A kleiner als alle Elemente in $\bar{A}$ sind).

Phase 1 läuft, solange $|C(x)| \geq r$. Ein Knoten auf einem Pfad in T von der Wurzel, bei dem $|C(x)|$ erstmals $= r$ wird, heißt kritisch. Jeder Pfad in T von der Wurzel zu einem Blatt enthält genau einen kritischen Knoten.

Beweis (Forts.):

Betrachte in T_A einen Pfad von der Wurzel zu einem kritischen Knoten x. Sei y ein Knoten auf diesem Pfad, z sein Kind. Es gilt:

$$|C(z)| + |c(z)| \geq |C(y)| + |c(y)| - 1$$



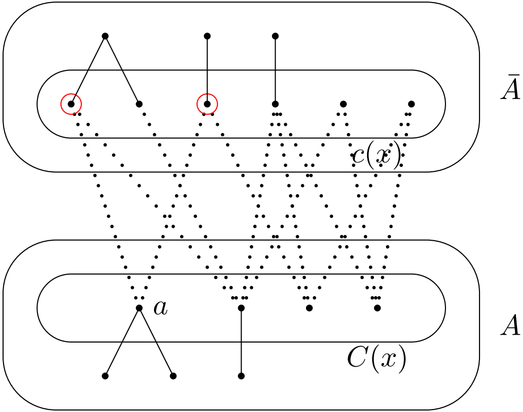
Da $|C(\text{Wurzel})| = |A| = i$ und $|c(\text{Wurzel})| = |\bar{A}| = n - i$, müssen überhalb eines jeden kritischen Knoten x mindestens

$$i - |C(x)| + n - i - |c(x)| = n - r - |c(x)|$$

Vergleiche erfolgen. Von jedem kritischen Knoten abwärts arbeitet der Gegenspieler nach einer Strategie für Phase 2. Sei x ein solcher kritischer Knoten. Dann ist $|C(x)| = r$.

Beweis (Forts.):

Phase 2: Sei $a \in C(x)$ ein Element mit minimalem $s(x, a)$.



$\#\odot = s(x, a)$
für $a \in A$

Beweis (Forts.):

Fall 1: $s(x,a) \geq s$. Betrachte irgendeinen Pfad von der Wurzel durch x zu einem Blatt w. Jeder solche Pfad muss mindestens $n-1$ Vergleiche enthalten, um answer(w) zu verifizieren: $\geq n-i$, für die answer(w) sich (direkt oder indirekt) als das kleinere Element ergibt, und $\geq i-1$, wo es sich als das größere ergibt. Damit sind $\geq (r-1)s$ Vergleiche redundant (nämlich alle die, die zwischen Elementen aus $C(x) \setminus \{\text{answer}(w)\}$ und Elementen in $\bar{A}$ erfolgt sind). Also:

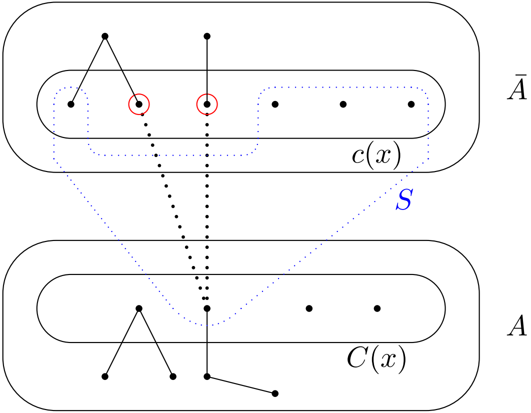
$$\begin{aligned}
\text{Höhe}(T) &\geq n-1+s(r-1) = n-r-s-1+(s+1)r \\
&= n-r-s+1+1+\log\left[\frac{\binom{n}{i}}{n-i+1}\right] \\
&> \log\left[\binom{n}{i}\frac{2^{n-p}}{n-i+1}\right].
\end{aligned}$$

In diesem Fall folgt also die Behauptung direkt!

Beweis (Forts.):

Fall 2: $s(x, a) < s$.

Fall 2.1: $|c(x)| \geq p$.



$\#\odot = s(x, a)$ für $a \in A$

Beweis (Forts.):

Sei $S := c(x) \cup \{a\} \backslash$ die Elemente, die in s(x, a) gezählt werden. Der Gegenspieler antwortet nun so, dass das Ergebnis das kleinste Element in S wird. Ist w ein Blatt in T_A unter x, so ist little(w)= $A - \{a\}$. Der Entscheidungsbaum T wird also gemäß folgender Regeln gestutzt (sei y der aktuelle Knoten und seien e und f die beiden Elemente, die in y verglichen werden):

2.1: falls $e, f \in S$, dann bleibt y erhalten

2.2: andernfalls sei o.B.d.A. $e \in A \setminus S$ oder $f \in \bar{A} \setminus S$; ersetze y mit seinem Unterbaum durch das Kind von y und dessen Unterbaum, das der Antwort auf $e < f$ entspricht.

Da überhalb des kritischen Knoten x keine zwei Elemente in S verglichen wurden, muss der Unterbaum von T_A unterhalb von x Tiefe $\geq |S| - 1$ haben.

Beweis (Forts.):

Zusammen mit Phase 1 ergibt sich eine Tiefe von T_A:

$$\begin{aligned}
&\geq n - r - |c(x)| + |S| - 1\\
&\geq n - r - |c(x)| + |c(x)| + 1 - (s-1) - 1\\
&= n - r - s + 1\\
&= n - p
\end{aligned}$$

Beweis (Forts.):

Fall 2.2: $|c(x)| < p$. Sei $S := C(x)$.
Die Regeln für Phase 2 sind in diesem Fall so, dass der Algorithmus als Antwort das Maximum von S bestimmt. Damit ergibt sich für die Tiefe von T_A:

$$\begin{aligned}
&\geq n - r - |c(x)| + |S| - 1 \\
&\geq n - r - (p-1) + r - 1 \\
&= n - p\,.
\end{aligned}$$

Beweis (Forts.):

Insgesamt ergibt sich also: Jeder Pfad in T_A von x zu einem Blatt hat mindestens die Länge $n-p$. Also enthält jedes T_A mindestens 2^{n-p} Blätter (von T).

Alle T_A's zusammen enthalten $\geq \binom{n}{i} 2^{n-p}$ Blätter von T, wobei jedes Blatt von T höchstens $n-i+1$ mal vorkommt: Sei w Blatt von T, dann ist little(w) eindeutig bestimmt und es muss, falls T_A das Blatt w enthält, little$(w) \subseteq A$ sein.

Für das Element in $A \setminus$ little(w) gibt es $\leq n-i+1$ Möglichkeiten. Damit ist die Anzahl der Blätter von $T \geq \frac{1}{n-i+1}\binom{n}{i}2^{n-p}$ und

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

$\square$

Literatur

L. Hyafil:
Bounds for selection
SIAM J. Comput. **5**, pp. 109–114 (1976)

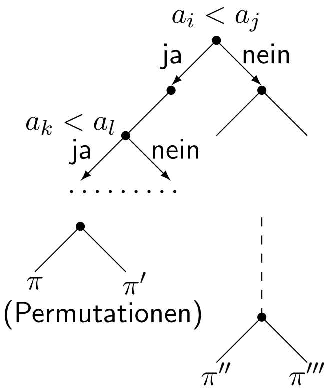
Sam Bent, John W. John:
Finding the median requires $2n$ comparisons
Proc. 17th Annual ACM Symposium on Theory of Computing, pp. 213–216 (1985)

John Welliaveetil John:
A new lower bound for the set-partitioning problem
SIAM J. Comput. **17**, pp. 640–647 (1988)

Dorit Dor, Uri Zwick:
Selecting the median
SIAM J. Comput. **28**(5), pp. 1722–1758 (1999)

7. Untere Schranke für (vergleichsbasiertes) Sortieren

Gegeben n Schlüssel, Queries $a_i \overset{?}{<} a_j$. Dies entspricht einem Entscheidungsbaum:



Beobachtung: Jede Permutation $\pi \in S_n$ kommt in mindestens einem Blatt des Entscheidungsbaums vor. Da $|S_n| = n!$, folgt, dass die Tiefe des Entscheidungsbaums

$$\geq \log_2 n!\,.$$

Stirling'sche Approximation: $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

Also

$$\begin{aligned}\text{Tiefe } \geq \log_2\left[\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right] &= n\log_2 n - n\log_2 e + \frac{1}{2}\log_2(2\pi n)\\ &= n\log_2 n - \mathcal{O}(n)\,.\end{aligned}$$

Satz 91

Jeder vergleichsbasierte Sortieralgorithmus benötigt für das Sortieren von n Schlüsseln mindestens

$$n \log_2 n - \mathcal{O}(n)$$

Vergleiche.

8. Bucketsort im Schnitt

Gegeben seien n zufällig und gleichverteilt gewählte Schlüssel im Intervall $]0, 1]$. Um diese zu sortieren:

1. Initialisiere Buckets $b_1, \ldots, b_n$
2. Lege a_i in Bucket $\lceil n \cdot a_i \rceil$
3. Sortiere Schlüssel innerhalb eines jeden Buckets
4. Konkateniere die Buckets

Satz 92

Wird zum Sortieren ein Sortieralgorithmus mit einer Zeitkomplexität von $\mathcal{O}(n^2)$ verwendet, dann hat obiger Bucketsort-Algorithmus im Durchschnitt eine lineare Laufzeit.

Beweis:

Sei n_i die Anzahl der Elemente im Bucket b_i (nach Schritt 2). Die Schritte 1,2 und 4 benötigen zusammen Zeit $\mathcal{O}(n)$. Die Zeit für Schritt 3 beträgt:

$$\mathcal{O}(\sum_{i=1}^{n} n_i^2)\,.$$

Die Erwartungswert für $\sum n_i^2$ lässt sich folgendermaßen abschätzen:

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{n} n_i^2\right] &\leq c \sum_{1\leq j<k\leq n} \Pr[a_j \text{ und } a_k \text{ enden im gleichen Bucket}]\\ &= c\left(\sum_{1\leq j<k\leq n} \frac{1}{n}\right) = \mathcal{O}(n)\,.\end{aligned}$$

$\square$