Technische Universität München
Institut für Informatik

Effiziente Algorithmen und Datenstrukturen
WS 1997/98 - Kapitel 2

Stand: 10. März 1998

Dozent:

Prof. Dr. Ernst W. Mayr

Mitschrift:

Martin Stumpf
Thomas Schöpf
Martin Hans
Guido Wimmel

Inhaltsverzeichnis

Kapitel 2

Selektion

2.1 Einleitung

Gegeben: Menge S von n Elementen aus einem total geordneten Universum U, $i \in \mathbb{N}$, $1 \le i \le n$.
Gesucht: i-kleinstes Element in S
Bekannt: Fälle $i = 1,\ n$ (Minimum, Maximum bestimmen)
Standardalgorithmus: $n-1$ Vergleiche
Zeigen: Diese Anzahl ist optimal:
Beweis:
Interpretiere jeden Algorithmus als Turnier. Ein Spiel wird vom kleineren Element gewonnen. Beobachtung: Jedes Element außer dem Gesamtsieger muß mindestens ein Spiel verloren haben $\Rightarrow n-1$ Vergleiche notwendig.
Nun: Bestimmung des Vize-Meisters (bzw. zweitkleinstes Element) $\square$

Satz: Das zweitkleinste Element kann mit $n + \lceil \log_2 n \rceil - 2$ Vergleichen bestimmt werden.

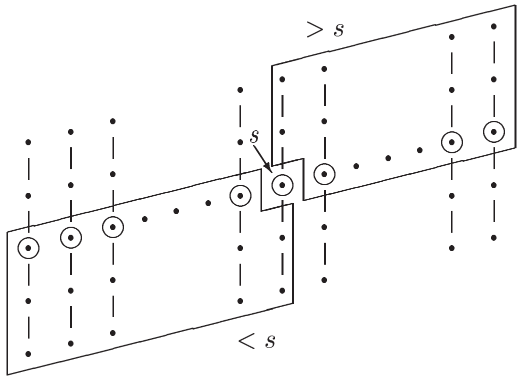
Beweis:
Bestimme das Minimum in einem K.O. Turnier: $(n-1)$ Vergleiche. Beobachtung: Das zweitkleinste Element ist unter den „Verlierern“ gegen das Minimum zu suchen. Deren Anzahl ist $\lceil \log_2 n \rceil$. Bestimme unter diesen Elementen wiederum das Minimum $\rightsquigarrow$ zweitkleinstes Element $(\lceil \log_2 n \rceil - 1)$ $\square$

2.2 Der Blum-Floyd-Pratt-Rivest-Tarjan Selektions-Algorithmus

Sei m eine kleine ungerade Zahl (etwa $5 \le m \le 15$). Wir betrachten folgenden Algorithmus BFPRT:

1. Teile S in $\left\lceil \frac{n}{m} \right\rceil$ Blöcke auf, $\left\lfloor \frac{n}{m} \right\rfloor$ davon mit m Elementen
2. Sortiere diese $\left\lfloor \frac{n}{m} \right\rfloor$ Blöcke; falls $\left\lfloor \frac{n}{m} \right\rfloor$ gerade, sortiere auch die restlichen Elemente
3. Sei S' die Menge der Mediane dieser sortierten Mengen. Bestimme rekursiv den Median dieser Mediane (also das $\left\lceil \frac{|S'|}{2} \right\rceil$-kleinste Element). Nenne diesen Median s.

4. Partitioniere $S - \{s\}$ in $S_1 := \{x \in S : x < s\}$, $S_2 := \{x \in S : x > s\}$.
   Bemerkung.: $|S_1|,\ |S_2| \geq \frac{n}{4}$, falls $n \geq 3m - 1$



5. Falls $i \leq |S_1|$, bestimme rekursiv das i-kleinste Element in S_1
   Falls $|S_2| > n - i$, bestimme rekursiv das $(i - |S_1| - 1)$-kleinste Element in S_2
   Ansonsten: return(s)

Komplexitätsbetrachtung (# Vergleiche im worst-case):

Sei $\mathrm{T}(n)$ die worst-case Anzahl von Vergleichen für $|S| = n$ des Algorithmus BFPRT. Es gilt:

$$\mathrm{T}(n) \leq \underbrace{\mathrm{T}\left(\left\lceil\frac{n}{m}\right\rceil\right)}_{3.} + \underbrace{\mathrm{T}\left(\left\lfloor\frac{3}{4}n\right\rfloor\right)}_{5.} + \underbrace{\left\lceil\frac{n}{m}\right\rceil}_{2.} C_m + \underbrace{\left\lfloor\frac{n}{2}\right\rfloor}_{4.}$$

(C_m ist die Anzahl von Vergleichen, um m Elemente zu sortieren, z.B. $C_5 = 7,\ C_{11} = 26$)

Satz: Der Selektions-Algorithmus BFPRT bestimmt das i-te Element von n Elementen mit $\mathcal{O}(n)$ Vergleichen (und Zeit).

Behauptung:

$$\mathrm{T}(n) \leq c_m n;$$

Behauptung ist ok, falls

$$\mathrm{T}\left(\left\lceil\frac{n}{m}\right\rceil\right) + \mathrm{T}\left(\left\lfloor\frac{3}{4}n\right\rfloor\right) + \left\lceil\frac{n}{m}\right\rceil C_m + \left\lfloor\frac{n}{2}\right\rfloor \leq c_m n, \text{ dies gilt, falls}$$

$$\left\lceil\frac{n}{m}\right\rceil c_m + \left\lfloor\frac{3}{4}n\right\rfloor c_m + \left\lceil\frac{n}{m}\right\rceil C_m + \left\lfloor\frac{n}{2}\right\rfloor \leq c_m n \qquad |\cdot \frac{1}{n}$$

$$\Leftrightarrow \text{(bis auf } \lceil\ \rceil, \lfloor\ \rfloor\text{)}\ \frac{c_m}{m} + \frac{3}{4}c_m + \frac{C_m}{m} + \frac{1}{2} \leq c_m$$

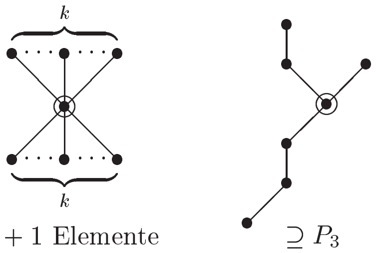
$$\Leftrightarrow -\frac{c_m}{m} - \frac{3}{4}c_m + c_m \geq \frac{C_m}{m} + \frac{1}{2}$$

$$\Leftrightarrow c_m \geq \frac{\frac{C_m}{m} + \frac{1}{2}}{1 - \frac{3}{4} - \frac{1}{m}}$$

Schlußbemerkung: $m = 11 \rightsquigarrow c_m \approx 20$

2.3 Der Schönhage-Paterson-Pippenger-Median-Algorithmus

Definition: Sei $k \in \mathbb{N}\backslash\{0\}$. P_k ist die folgende partielle Ordnung:

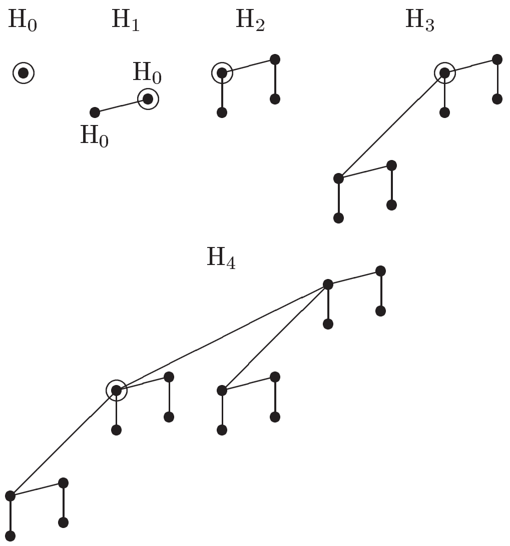


$2k+1$ Elemente $\qquad \supseteq P_3$

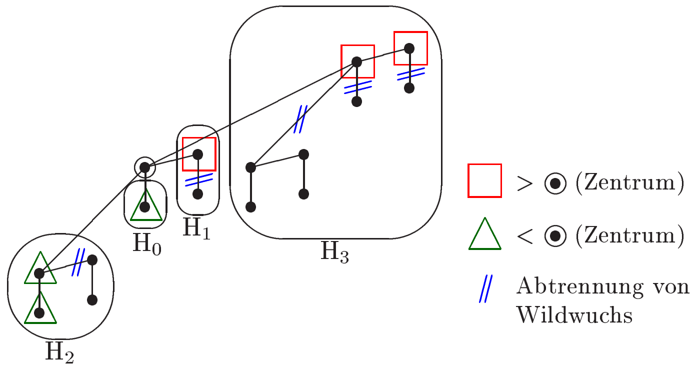
Spezielle Binomialbäume mit „Zentren“.

Definition:

1. Der Baum H_0 besteht aus einem Knoten und dieser ist auch das Zentrum.
2. $H_{2h} (h > 0)$ besteht aus zwei H_{2h-1}, deren Zentren durch eine neue Kante verbunden sind. Das Zentrum des H_{2h} ist das kleinere der beiden Zentren der H_{2h-1}.
3. $H_{2h+1} (h \geq 0)$ besteht aus zwei H_{2h}, deren Zentren durch eine neue Kante verbunden sind, sein Zentrum ist das größere dieser beiden Zentren.



Das Zerlegungslemma:



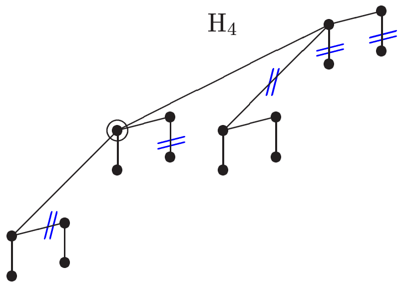
a) H_h hat 2^h Knoten, es werden $2^h - 1$ Vergleiche benötigt, um H_h zu konstruieren.

b) H_{2h} kann zerlegt werden in
   - sein Zentrum
   - eine Menge $\{H_1, H_3, \ldots, H_{2h-1}\}$ von disjunkten Teilbäumen, deren Zentren alle größer sind als das Zentrum von H_{2h}.
   - eine Menge $\{H_0, H_2, H_4, \ldots, H_{2h-2}\}$ von disjunkten Teilbäumen mit Zentren kleiner als das von H_{2h}.

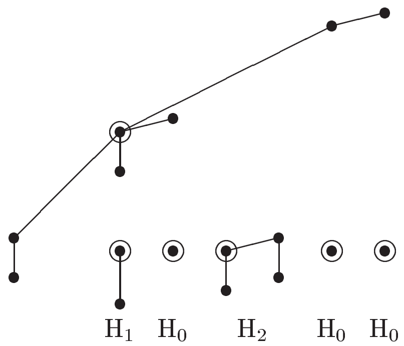
c) H_{2k+1} kann so zerlegt werden, daß die Zusammenhangskomponente des Zentrums genau 2^k Knoten $\geq$ dem Zentrum enthält, indem höchstens $2^{k+1} - 1$ Kanten entfernt werden.
   H_{2k} kann so zerlegt werden, daß die Zusammenhangskomponente des Zentrums genau 2^k Knoten enthält, die alle $\leq$ dem Zentrum sind, indem höchstens $2^k - 1$ Kanten entfernt werden.

d) Falls $k \leq 2^h - 1$, dann kann H_{2h} so zerlegt werden, daß die Zusammenhangskomponente des Zentrums genau $2k + 1$ Elemente enthält, von denen k größer und k kleiner als das Zentrum sind ($\Rightarrow P_k$).
   Dazu genügt es, höchstens $3k + 2h$ Kanten zu entfernen. Die restlichen Zusammenhangskomponenten sind wieder H_i.

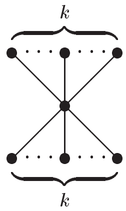
Bemerkung: Bei jedem Konstruktionsschritt wird ein Vergleich durchgeführt, um zu bestimmen, welcher der beiden Teilbäume das kleinere Zentrum hat. Im Algorithmus von Schönhage, Paterson und Pippenger werden aus Teilstücken H_r größere Bäume H_{r+1} zusammengebaut, wodurch schrittweise eine partielle Ordnung auf den Eingabewerten bestimmt wird. Wurde ein Baum H_{2h} hinreichender Größe hergestellt, so wird er durch Zerlegung in einen Baum umgewandelt, der nur noch sein altes Zentrum, sowie k darüberliegende und k darunterliegende Elemente enthält, wobei $k \leq 2^h - 1$. Im folgenden Beispiel wollen wir H_4 zerlegen und wählen $k = 3$:



Um einen H_4 derart zu zerlegen, müssen wir 5 Kanten aufbrechen. Dabei werden drei H_0, ein H_1 sowie ein H_2 abgespalten.



Übrig bleibt die gewünschte Struktur mit k Knoten über dem Zentrum und k unter dem Zentrum, wodurch eine partielle Ordnung auf $2k+1$ Eingabewerten bestimmt wurde:



$2k + 1$ Elemente

Die bei der Zerlegung angefallenen Reststücke werden beim Aufbau weiterer Bäume benutzt. So geht das bereits angesammelte Wissen über die Ordnung der Elemente nicht verloren.

Beweis der Teile a) bis d) des Zerlegungslemmas:

a) **Lemma 1:** H_r hat 2^r Knoten, es werden $2^r - 1$ Vergleiche benötigt, um H_r aufzubauen.

Beweis: In jedem der r Konstruktionsschritte wird die Anzahl der Knoten verdoppelt. Da wir mit einem Knoten beginnen, hat H_r folglich 2^r Knoten. Die Anzahl der notwendigen Vergleiche C_r unterliegt folgender Rekursionsgleichung ($r \geq 1$): $C_r = 1 + 2C_{r-1}$ und $C_0 = 0$. Damit folgt sofort $C_r = 2^r - 1$. □

b) **Lemma 2:** H_r kann in folgende disjunkte Bereiche unterteilt werden:

- sein Zentrum,
- eine Reihe H_1, H_3, ..., H_{r-1} (r gerade) bzw. ..., H_{r-2} (r ungerade) von Unterbäumen, deren Zentren über dem von H_r liegen,
- eine Reihe H_0, H_2, ..., H_{r-2} (r gerade) bzw. ..., H_{r-1} (r ungerade) von Unterbäumen, deren Zentren unter dem von H_r liegen.

Beweis: durch Induktion über r. Induktionsanfang: für H_0 gilt die Behauptung. Induktionsannahme: die Behauptung gelte für H_{r-1}.

1. Sei $r = 2h, h > 0$. H_{2h} besteht aus zwei H_{2h-1}, wobei das kleinere der beiden alten Zentren das neue Zentrum z bildet. Wende auf den H_{2h-1}, der z enthält, die Induktionsannahme an. Wir können diesen Unterbaum also in z, sowie $H_1, H_3, ..., H_{2h-3}$ (Zentren über z) und $H_0, H_2, ..., H_{2h-2}$ (Zentren unter z) partitionieren. Zusammen mit dem H_{2h-1}, dessen Zentrum über z liegt, ergibt sich die Induktionsbehauptung für H_{2h}.
2. Sei $r = 2h + 1, h \geq 0$. H_{2h+1} besteht aus zwei H_{2h}, wobei das größere der beiden alten Zentren das neue Zentrum z bildet. Wende auf den H_{2h}, der z enthält, die Induktionsannahme an. Wir können diesen Unterbaum also in z, sowie H_1, H_3, ..., H_{2h-1} (Zentren über z) und H_0, H_2, ..., H_{2h-2} (Zentren unter z) partitionieren. Zusammen mit dem H_{2h}, dessen Zentrum unter z liegt, ergibt sich die Induktionsbehauptung fr H_{2h+1}.

□

Wir bezeichnen im folgenden mit H_{2h}^- den Baum, der entsteht, wenn wir H_{2h} so zerlegen, da alle Elemente oberhalb des Zentrums wegfallen. Mit H_{2h+1}^+ bezeichnen wir den Baum, der entsteht, wenn wir H_{2h+1} so zerlegen, da alle Elemente unterhalb des Zentrums wegfallen.

c) **Lemma 3:** H^-_{2h} und H^+_{2h+1} haben jeweils 2^h Knoten. Bei der Herstellung aus H_{2h} bzw. H_{2h+1} werden $2^h - 1$ bzw. $2^{h+1} - 1$ Kanten aufgebrochen. Die wegfallenden Teile haben die Form $H_s, s < 2h$ bzw. $s < 2h + 1$.

Beweis: durch Induktion über r. Induktionsanfang: für H_0 und H_1 gilt die Behauptung. Induktionsannahme: die Behauptung gilt für alle $H_p, p < r$.

1. Sei $r = 2h, h > 0$. Wir betrachten die Partitionierung von H_{2h} mit Zentrum z wie in Lemma 2. Die Unterbäume H_1, H_3, ..., H_{2h-1} haben ihre Zentren oberhalb von z. Wir trennen sie von H_{2h}, indem wir h Kanten aufbrechen. Die abgetrennten Teile haben offensichtlich die Form $H_s, s < 2h$. Bei den Unterbäumen H_0, H_2, ..., H_{2h-2}, mit Zentren unterhalb von z, wenden wir jeweils die Induktionsannahme an, d.h. wir erzeugen H^-_0, H^-_2, ..., H^-_{2h-2}. Als Ergebnis erhalten wir H^-_{2h}. Damit gilt für die Zahl der aufzubrechenden Kanten $K^-(2h)$ zur Herstellung von H^-_{2h}:

$$K^-(2h) = h + \sum_{i=0}^{h-1} K^-(2i) \overset{I.A.}{=} h + \sum_{i=0}^{h-1}(2^i - 1) = \sum_{i=0}^{h-1} 2^i = 2^h - 1.$$

Für die Zahl $E^-(2h)$ der Elemente in H^-_{2h} gilt:

$$E^-(2h) = 1 + \sum_{i=0}^{h-1} E^-(2i) \overset{I.A.}{=} 1 + \sum_{i=0}^{h-1} 2^i = 1 + \underbrace{\sum_{i=1}^{h} 2^{i-1}}_{2^h-1} = 2^h.$$

2. Sei $r = 2h+1, h > 0$. Wir betrachten die Partitionierung von H_{2h+1} mit Zentrum z wie in Lemma 2. Die Unterbäume H_0, H_2, ..., H_{2h} haben ihre Zentren unterhalb von z. Wir trennen sie von H_{2h+1}, indem wir $h + 1$ Kanten aufbrechen. Die abgetrennten Teile haben offensichtlich die Form $H_s, s < 2h + 1$. Bei den Unterbäumen H_1, H_3, ..., H_{2h-1}, mit Zentren oberhalb von z, wenden wir jeweils die Induktionsannahme an, d.h. wir erzeugen H^+_1, H^+_3, ..., H^+_{2h-1}. Als Ergebnis erhalten wir H^+_{2h+1}. Damit gilt für die Zahl der aufzubrechenden Kanten $K^+(2h+1)$ zur Herstellung von H^+_{2h+1}:

$$\begin{aligned} K^+(2h+1) &= h + 1 + \sum_{i=1}^{h} K^+(2(i-1)+1) \\ &\overset{I.A.}{=} h + 1 + \sum_{i=1}^{h}(2^i - 1) = 1 + \sum_{i=1}^{h} 2^i \\ &= 1 + \underbrace{\sum_{i=1}^{h+1} 2^{i-1}}_{2^{h+1}-1} - 1 = 2^{h+1} - 1. \end{aligned}$$

Für die Zahl $E^+(2h+1)$ der Elemente in H^+_{2h+1} gilt:

$$E^+(2h+1) = 1 + \sum_{i=1}^{h} E^+(2(i-1)+1) \overset{I.A.}{=} 1 + \underbrace{\sum_{i=1}^{h} 2^{i-1}}_{2^h-1} = 2^h.$$

□

d) **Lemma 4:** Falls $k \leq 2^h - 1$, dann kann H_{2h} so zerlegt werden, daß die Komponente des Zentrums genau $2k+1$ Elemente enthält, k davon über und k unter dem Zentrum. Dazu müssen $\leq 3k + 2h$ Kanten entfernt werden. Die entfernten Teile sind von der Form $H_s, s < 2h$.

Beweis: Betrachte die Binärdarstellung von $k = k_0 2^0 + k_1 2^1 + ... + k_{h-1} 2^{h-1}$ und die Partitionierung von H_{2h} mit Zentrum z wie in Lemma 2.

- Für jedes i mit $k_i = 1$, betrachte H_{2i+1} aus der Sequenz H_1, H_3, ..., H_{2h-1} von Unterbäumen deren Zentren oberhalb von z liegen und schneide alle Elemente aus H_{2i+1}, die kleiner als sein Zentrum sind (bilde also H^+_{2i+1}). Dazu müssen höchstens $2k$ Kanten aufgebrochen werden, denn jedes $k_i = 1$ steht für 2^i in k, kostet uns aber nach Lemma 2 $K^+(2i+1) = 2^{i+1} - 1$ Kanten, also:

  $$\sum_{i=0}^{h-1} k_i K^+(2i+1) \leq 2k.$$

  Für jedes i mit $k_i = 0$, schneide H_{2i+1} ganz weg. Dabei werden höchstens h Kanten aufgebrochen. Genau k Elemente oberhalb von z bleiben zurück, da jedes $k_i = 1$ für 2^i in k steht, und ein H^+_{2i+1} genau $E^+(2i+1) = 2^i$ Elemente enthält, also:

  $$\sum_{i=0}^{h-1} k_i E^+(2i+1) = k.$$

- Für jedes i mit $k_i = 1$, betrachte H_{2i} aus der Sequenz H_0, H_2, ..., H_{2h-2} von Unterbäumen, deren Zentren unterhalb von z liegen und schneide alle Elemente aus H_{2i}, die größer als sein Zentrum sind (bilde also H^-_{2i}). Dazu müssen höchstens $k-1$ Kanten aufgebrochen werden, denn jedes $k_i = 1$ steht für 2^i in k und kostet uns nach Lemma 3 $K^-(2i) = 2^i - 1$ Kanten, also:

  $$\sum_{i=0}^{h-1} k_i (2^i - 1) \leq k - 1.$$

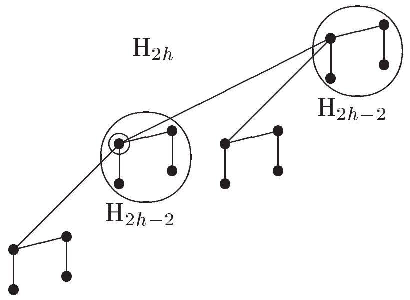
  Für jedes i mit $k_i = 0$, schneide H_{2i} ganz weg. Dabei werden höchstens h Kanten aufgebrochen. Genau k Elemente unterhalb von z bleiben zurück, da jedes $k_i = 1$ für 2^i in k steht, und ein H^-_{2i} genau $E^-(2i) = 2^i$ Elemente enthält, also:

  $$\sum_{i=0}^{h-1} k_i E^-(2i) = k.$$

Damit ergibt sich fr die Gesamtanzahl aufzubrechender Kanten eine obere Schranke von $3k + 2h$. Lemma 3 liefert uns darüberhinaus die gewünschte Aussage ber die Form der abgetrennten Teile. □

Betrachte H_{2h}

- „größer“: $H_{2h-1}, H_{2h-3}, \ldots, H_1$
- „kleiner“: $H_{2h-2}, H_{2h-4}, \ldots, H_0$

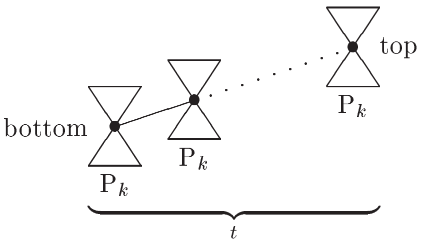


U(h) := Anzahl der Elemente in H_{2h} $\geq$ Zentrum: $\mathrm{U}(h) = 2\mathrm{U}(h-1) = 2^h\mathrm{U}(0) = 2^h$
D(h) := Anzahl der Elemente in H_{2h} $\leq$ Zentrum: $\mathrm{D}(h) = 2\mathrm{D}(h-1) = 2^h\mathrm{D}(0) = 2^h$

Anzahl der Kanten, die entfernt werden müssen:

$$\left.\begin{aligned} \mathrm{C}_u(h) &\leq 2 + 2\mathrm{C}_u(h-1) \\ &= 2 + 4 + 2^3 + \ldots + 2^h \\ &= 2^{h+1} - 2 \\ \mathrm{C}_d(h) &\leq 1 + 2\mathrm{C}_d(h-1) \\ &= 2^h - 1 \end{aligned}\right\} \quad \mathrm{C}_H \leq 2^{h+1} - 2 + 2^h - 1 \approx 3 \cdot 2^h$$

Kette von P_k's:



Gesamtzahl der Elemente:

n

$t(2k+1)$ in den P_k's

$r = n - t(2k+1)$ Rest

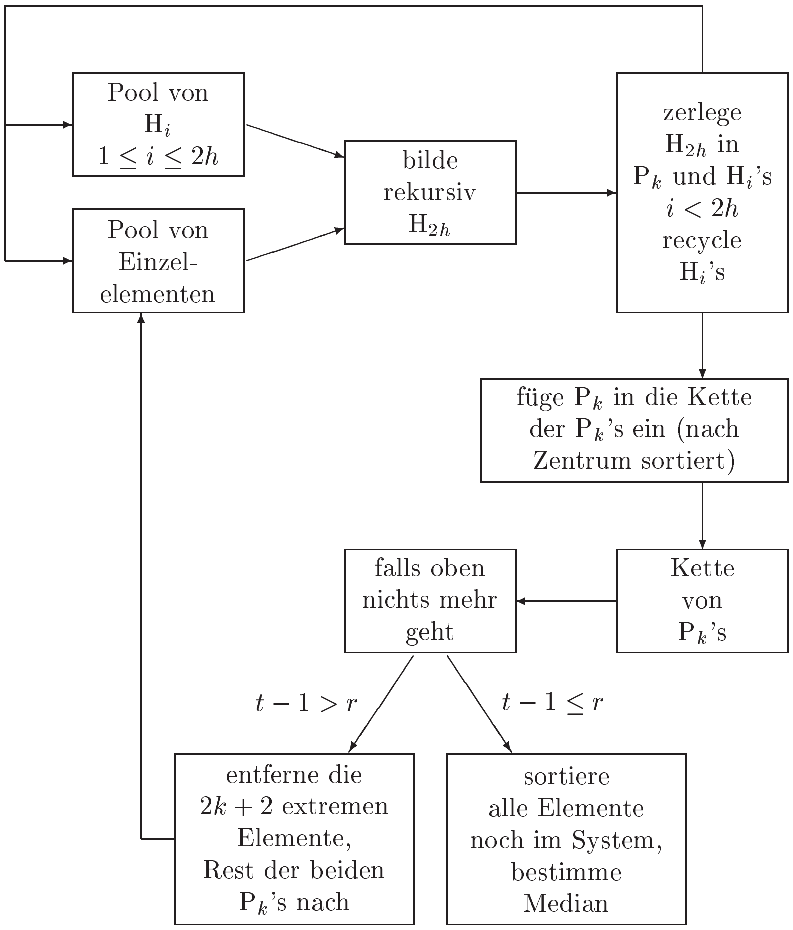
Wenn $r < t - 1$, dann wissen wir, daß top größer ist als

$$k + (t-1)(k+1) > k + (k+1)\left(\frac{n+1}{2k+2} - 1\right) = \frac{n-1}{2}$$

$\Rightarrow$ top $>$ Median

$$k := \left\lfloor n^{\frac{1}{4}} \right\rfloor$$

h sdg. $2^{h-1} \leq k < 2^h$



Definiere: $r :=$ Anzahl der noch im H_{2h} Produktionsprozeß steckenden Elemente. (für jedes $i < 2h$ höchstens ein $H_i \Rightarrow r \leq \sum_{i=0}^{2h-1} = 2^{2h} - 1$)
$R :=$ Anzahl der im letzten Schritt zu sortierenden Elemente. Es gilt: $t \leq r + 1$, und damit

$$R = t(2k+1) + r \leq 2^{2h}(2k+1) + 2^{2h} - 1$$

$m :=$ Gesamtzahl der im Algorithmus produzierten P_k's.

$$m = t + 2\frac{n-R}{2(k+1)} = t + \frac{n-R}{k+1}$$

Gesamtzahl der vom Algorithmus durchgeführten Vergleiche =

1. Anzahl der Kanten in allen P_k's
2. + Anzahl der Kanten, die gelöscht werden, um die P_k's zu formen
3. + Anzahl der Kanten, die zum Schluß in übriggebliebenen H_i's ($i < 2h$) stecken
4. + Anzahl der Vergleiche, um jedes Zentrum der P_k's in die (sortierte) Kette einzufügen
5. + Anzahl der Vergleiche, um die zum Schluß übriggebliebenenen R Elemente zu sortieren

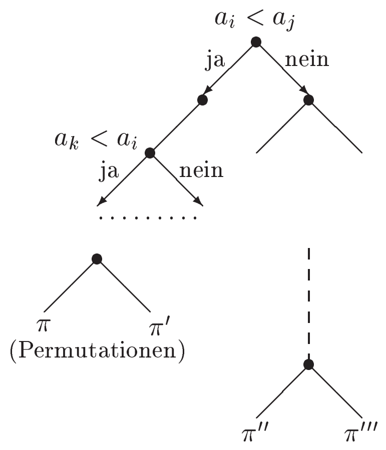
$$\leq \left(\frac{n-R}{k+1}+t\right)\left[\underbrace{2k}_{1}+\underbrace{3k+2h}_{2}+\underbrace{\log\frac{n}{2k+1}}_{4}\right]+\underbrace{R\log R}_{5}+\underbrace{r}_{3}$$

$k := \left\lfloor n^{\frac{1}{4}} \right\rfloor$, h so daß $2^{h-1} \leq k < 2^h$

$$\begin{aligned} \text{Damit } r &= \mathcal{O}(k^2) \\ t &= \mathcal{O}(k^2) \text{ zum Schluß} \\ R &= \mathcal{O}(k^3) \text{, und damit die} \\ \text{Anzahl der Vergleiche } &= \mathrm{T}(n) \leq 5n + o(n) \end{aligned}$$

Verbesserte Version: $T(n) = 3n + o(n)$
Dor/Zwick: $2,95n + o(n)$

Einschub: Untere Schranke für das vergleichsbasierte Sortieren
Entscheidungsbaum:



Stirling: $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$
Binärbaum, alle Permutationen $\in S_n$ kommen als Blattmarkierungen vor.

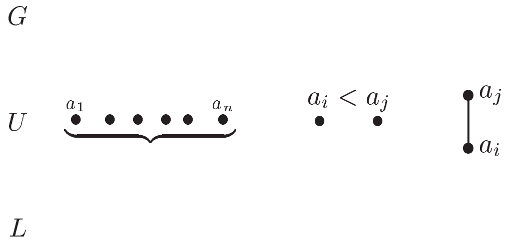
$$\begin{aligned} \text{worstcase Laufzeit} &\geq \text{Tiefe des Baums} \\ &\geq \log_2(\text{Anzahl der Blätter}) \\ &= \log_2(n!) \\ &= n \log n + \mathcal{O}(n) \end{aligned}$$

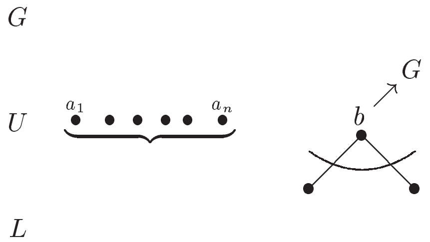
2.4 Eine untere Schranke für die Medianbestimmung

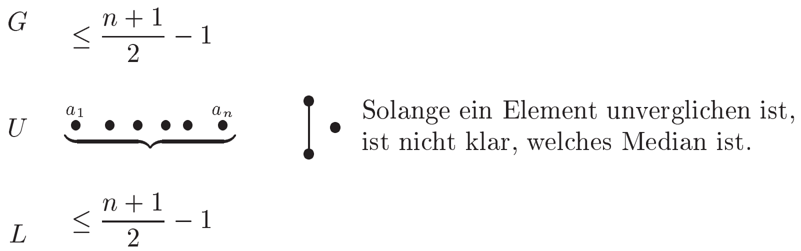
Satz: Jeder (vergleichsbasierte) Medianalgorithmus benötigt mindestens $\lceil \frac{3n}{2} \rceil - 2$ Vergleiche

Beweis: Gegenspielerargument
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. Der Algorithmus stellt nun Fragen der Form $x < y$. Der Gegenspieler gibt

konsistente Antworten, die den Algorithmus jedoch dazu zwingen, möglichst viele Fragen zu stellen (d.h. U soll möglichst ungeordnet bleiben).



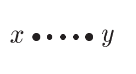
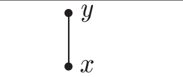
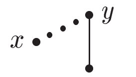
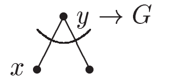
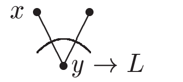
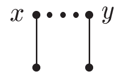
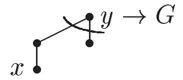
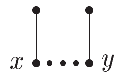
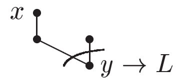
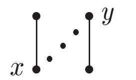
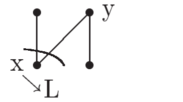




Solange $|L|,\ |G| < \frac{n+1}{2}$, kann der Algorithmus annehmen, daß der Median in U ist. Solange U mindestens zwei unverglichene Elemente und keine Zusammenhangskomponente mit > 2 Elementen enthält, kann der Algorithmus den Median nicht bestimmen.
Strategie des Gegenspielers: Zwei Phasen
Erste Phase: Fragen des Algorithmus: $\bullet \ldots \bullet$

$C :=$ Anzahl der Vergleiche
$P :=$ Anzahl der Paare in U
Dann gilt folgende Invariante: $C - P + 2|U| \geq 2n$

Beweis: vollständige Induktion □

	Status	Antwort des Gegenspielers	Anzahl der Paare in U	$\lvert U\rvert$	$\lvert L\rvert$	$\lvert G\rvert$
1.			+1	—	—	—
2.			-1	-1	0	+1
3.			-1	-1	1	0
4.			-1	-1	0	+1
5.			-1	-1	+1	0
6.			-1	-1	+1	0

Die erste Phase endet, wenn $|L| = \frac{n-1}{2}$ oder $|G| = \frac{n-1}{2}$ oder wenn U keine unverglichenen Elemente mehr enthält, d.h. $|U| \leq 2$. Wenn eine dieser Bedingungen eintritt, beginnt der Gegenspieler mit Phase 2:

- 1. Fall:

  $$|U| \leq 2$$

  Dann folgt aus der Invariante für Phase 1:

  $$C - P + 2 \cdot 2 \geq 2n, \text{ also } C \geq 2n - 3$$

- 2. Fall: $|L|$ wird $\frac{n-1}{2}$ oder $|G|$ 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 zu bestimmen (da dieses unter den Vorgaben der Median ist).
  Von den dazu nötigen $|U|-1$ Vergleichen hat der Algorithmus bereits höchstens P ausgeführt (P = Anzahl der Paare), bleiben also $|U| - 1 - P$.
  Zusammen (Phase 1 + Phase 2) ergeben sich damit $\geq C + |U| - 1 - P$ Vergleiche.
  Da $C \geq 2n + P - 2|U|$ ist die Anzahl der Vergleiche:

  $$\geq 2n + P - 2|U| + |U| - 1 - P = 2n - |U| - 1 \geq \left\lceil \frac{3n}{2} - 2 \right\rceil$$

  da $|U| \leq \frac{n-1}{2}$.

□

2.5 Eine bessere untere Schranke

Literatur: Samuel W. Bent, John W. John: “Finding the median requires $2n$ comparisons.” Proceedings of the 17th Annual ACM Symposium on Theory of Computing 1985, p.213–216.

Satz: 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, mit

$$p := 2\sqrt{\log\left[\binom{n}{i}\frac{1}{n-i+1}\right] + 3} - 2,$$
$$\text{Höhe}(T) \geq \log\left[\binom{n}{i}\frac{2^{n-p}}{n-i+1}\right]$$

($\rightarrow 2n - o(n)$ für $i = \frac{n}{2}$ (untere Schranke für Median))

Beweis: Gegenspielerargument + Abzählargument
„T hat viele Blätter“, A Teilmenge der Schlüssel, $|A| = i$
T_A Teilbaum von T, alle Blätter von T_A sind auch Blätter von T.
Zu zeigen: T_A hat viele Blätter. ($\binom{n}{i}$ Möglichkeiten, T_A’s zu konstruieren, Anzahl Blätter in T_A: 2^{n-p}, jedes Blatt von T kommt in höchstens $n-i+1$ T_A’s vor)
Beobachtungen: Jedes Blatt w von T liefert folgende Informationen:

answer(w)	$\xrightarrow{\text{liefert}}$	i-kleinstes Element x
little(w)	$\xrightarrow{\text{liefert}}$	$i-1$ Elemente $< x$
big(w)	$\xrightarrow{\text{liefert}}$	$n-i$ Elemente $> x$

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:

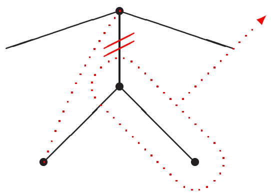
- T_A ist Binärbaum
- jedes Blatt von T_A ist auch Blatt von T (keine zus. Blätter durch Abschneiden)
- für jedes Blatt w von T_A gilt: little(w) $\subseteq A$

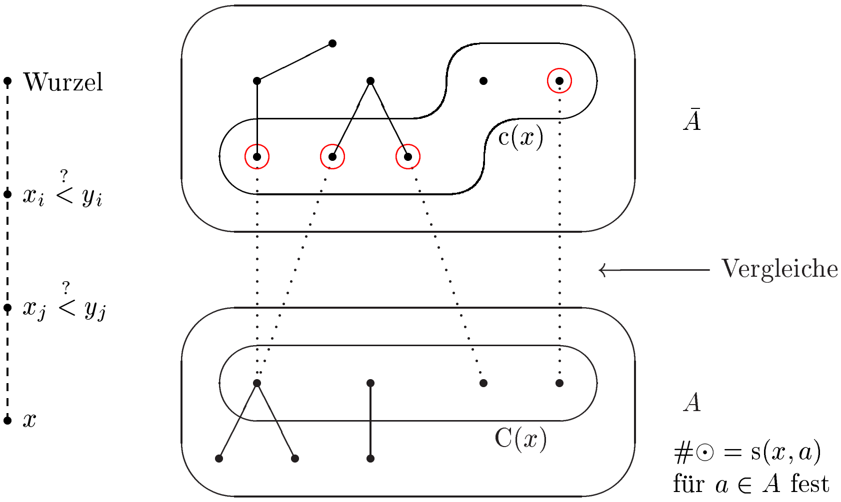
Setze $\bar{A}$ das Komplement von A,

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

Es gilt: $p = r + s - 1$

Das Zurechtstutzen von T erfolgt in zwei Phasen:





Erste Phase von der Wurzel nach unten (Breitensuche)
Betrachte Knoten x. Definiere:

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

Regeln für Phase 1:
Es werden in Knoten x die Elemente a und b verglichen:

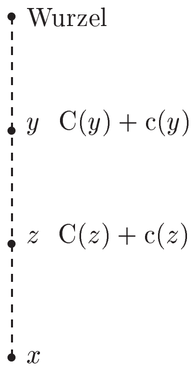
Regel 1.1: Falls $a \in A$ und $b \in A$ behalte x in T_A bei.

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

Regel 1.3: Sei o.B.d.A. $b \in \bar{A}$, $a \in A$. Ersetze den Unterbaum in T mit Wurzel x mit dem Unterbaum, dessen Wurzel das Kind von x ist, das dem Ergebnis entspricht (d.h. lasse den Vergleich $a < b$ aus, nach Voraussetzung $A < \bar{A}$).

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

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



Dann gilt:

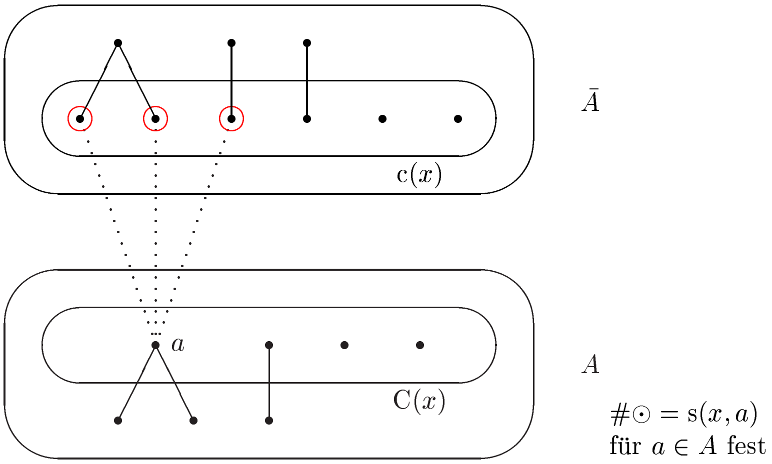
$$|\mathrm{C}(z)| + |\mathrm{c}(z)| \geq |\mathrm{C}(y)| + |\mathrm{c}(y)| - 1$$

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

$$i - |\mathrm{C}(x)| + n - i - |\mathrm{c}(x)| = n - r - |\mathrm{c}(x)|$$

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

Sei $a \in \mathrm{C}(x)$ ein Element mit minimalem $\mathrm{s}(x,a)$.



Fall 1: $\mathrm{s}(x,a) \geq s$.

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 $\mathrm{C}(x) \setminus \{\mathrm{answer}(w)\}$ und Elementen in $\bar{A}$ erfolgt sind).

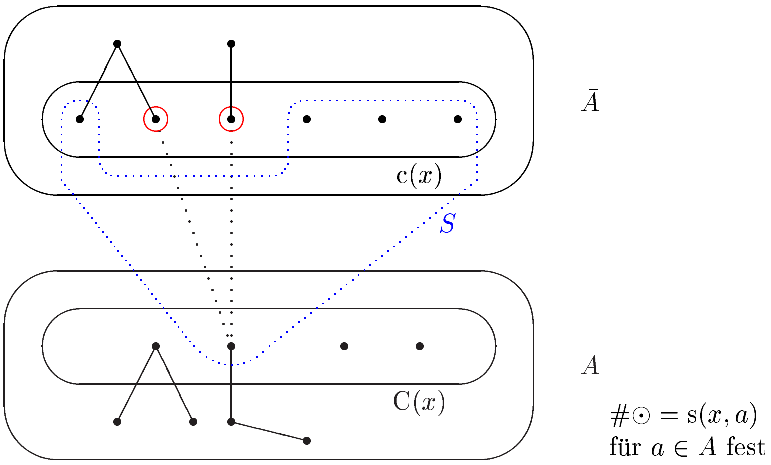
Also:

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

In diesem Fall folgt also die Behauptung direkt!

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

Fall 2.1:



$|c(x)| \geq p$. Sei $S := c(x) \cup \{a\} \backslash$Elemente, die in $s(x, a)$ gezählt werden. Der Gegenspieler antwortet nun so, daß das Ergebnis das kleinste Element in S wird.
Der Entscheidungsbaum T wird also gemäß folgender Regeln gestutzt (sei y der aktuelle Knoten und seien a und b die beiden Elemente, die in y verglichen werden):

Regel 2.1: falls $a, b \in S$, dann bleibt y erhalten.

Regel 2.2: andernfalls, sei oBdA. $a \in A \setminus S$ oder $b \in \bar{A} \setminus S$; ersetze y mit seinem Unterbaum durch das Kind von y und dessen Unterbaum, das der Antwort von $a < b$ entspricht.

Also, 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.
Zusammen mit Phase 1 ergibt sich eine Tiefe von T_A:

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

Fall 2.2: $|c(x)| < p$. $S := C(x)$
Regeln so, daß der Algorithmus als Antwort das Maximum von S bestimmt. Tiefe von T_A:

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

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 muß, falls T_A w auch enthält, $\subseteq A$ sein. Es gibt $\leq n - i + 1$ solche A's. $\square$