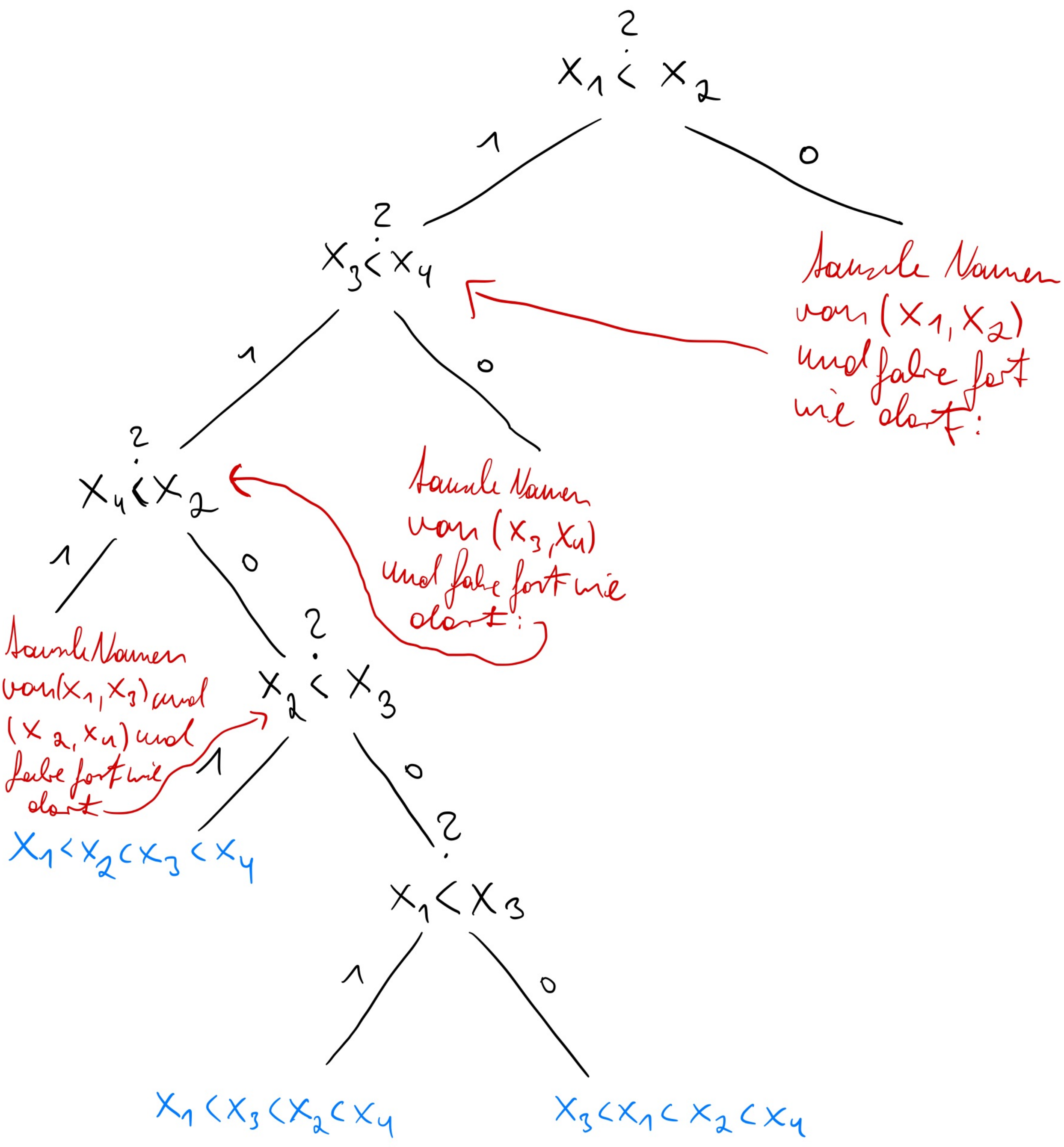
verschiedene?

Aufgabe 2 (Untere Schranken für Blackbox-Verfahren) (5 Punkte)

Wir betrachten Blackboxverfahren, die als Eingabe n paarweise natürliche Zahlen $x_1, \ldots, x_n$ erhalten und diese in aufsteigender Größe sortieren. Auf die Eingabe kann nur durch Fragen der Art „$x_i < x_j$?“ zugegriffen werden, diese werden mit ja oder nein beantwortet.

a) Geben Sie ein solches Verfahren an, dass für $n = 4$ nicht mehr als 5 Fragen benötigt. Stellen Sie Ihr Verfahren umgangssprachlich dar oder als binären Entscheidungsbaum, bei dem die Knoten mit Fragen wie oben beschriftet sind.



b) Zeigen Sie, dass für das Sortieren der Eingabe im schlechtesten Fall für $n = 4$ mindestens 5 und für $n = 5$ mindestens 7 Fragen nötig sind.

Ein Algorithmus der das Problem für $n \in \mathbb{N}$ ideal löst, muss so vorgehen, dass bei jeder Frage $x_i \overset{?}{<} x_j$ die Menge der auszuschließenden Permutationen in etwa gleich groß ist für beide Antworten. Ansonsten existiert immer ein Pfad bei dem weniger Möglichkeiten ausschlossen werden wie beim idealen Algorithmus, da die Summe der Möglichkeiten für $x_i < x_j$ und $x_j < x_i$ gleich der Anzahl der Möglichkeiten vor dem Stellen der Frage ist. Da für $n = 4$ insgesamt $4! = 24$ Permutationen möglich sind kann ein ideal arbeitender Algorithmus die Anzahl an Möglichkeiten nach 4 Schritten im schlechtesten Fall auf $\left\lceil \frac{24}{2^4} \right\rceil = 2$ beschränken.

Es sind also immer mindestens 5 Fragen nötig, da dann die Anzahl auf $\left\lceil \frac{24}{2^5} \right\rceil = 1$ reduziert und somit die Lösung gefunden wird.

Analog gilt für $n = 5$: $\left\lceil \frac{5!}{2^6} \right\rceil = \left\lceil \frac{120}{64} \right\rceil = 2$

aber $\left\lceil \frac{120}{2^7} \right\rceil = \left\lceil \frac{120}{128} \right\rceil = 1$

Also kann im schlechtesten Fall mit 6 Fragen die Anzahl an Möglichkeiten nur auf 2 reduziert werden, deshalb sind 7 Fragen mindestens nötig.

c) Gelten die unteren Schranken in Teil b noch, wenn als Frage auch beliebige Gleichungen und Ungleichungen zwischen arithmetischen Ausdrücken in den x_i wie „$x_1 + x_2 < x_4 - x_3$?“ oder „$x_1 = x_2 * x_3$?“ zugelassen sind und zusätzlich die x_i gleich den Zahlen 1 bis n sind?

Ja die Schranken gelten noch, denn wie bereits erwähnt können bei einem Algorithmus im schlechtesten Fall nach m Schritten nicht weniger als $\left\lceil \frac{n!}{2^m} \right\rceil$ Möglichkeiten übrig bleiben. Diese Feststellung ist unabhängig von den Operationen und welche Zahlen genau permutiert werden.

(A1) Wir bemerken:

Anton, Maximilian, Johannes, bei KAMAL

* Binomial braucht für $n=34$, $k=15$ 15,886 s.

* Binomial_fast funktioniert für $n > 12$ nicht mehr (falsche Ergebnisse), vermutlich liegt das daran, dass $13! > 2^{32}$ ist und die Zahl somit nicht mehr korrekt dargestellt werden kann (zumindest nicht vom Typ „int").

* Binomial funktioniert mindestens bis $n=34$ noch.

* Binomial_fast ist für $n < 12$ oft schneller als Binomial, aber nicht immer.

A3 z.z. $d_n(a \times b) = s_n(d_n(a) \times d_n(b))$ (*) falls $a, b, a \times b \in [-2^{n-1}, \ldots, 2^{n-1}-1]$

Wir zeigen vier Fälle, analog zum Beweis zur Addition

① $a, b \geq 0$ $\Rightarrow a \times b \geq 0$

nach Voraussetzung ist $a \times b < 2^{n-1}$ & $a, b < 2^{n-1}$

$$\Rightarrow s_n(d_n(a) \times d_n(b)) \underset{a,b<2^n-1}{=} s_n(a \times b) \underset{a \times b < 2^{n-1}}{=} a \times b \underset{a \times b < 2^{n-1}}{=} d_n(a \times b)$$

② Fall $a < 0, b \geq 0$ $\Rightarrow a \times b \leq 0$

nach Voraussetzung gilt dann

$$s_n(d_n(a) \times d_n(b)) = s_n((2^n + a) \times b) = s_n(2^n \times b + a \times b)$$

$$= (2^n \times b + a \times b) \bmod 2^n$$

$$= a \times b \bmod 2^n \underset{a \times b < 2^{n-1}}{=} a \times b = d_n(a \times b).$$

③ Fall $a \geq 0, b < 0$ analog zu Fall ②

④ Fall $a, b < 0$ $\Rightarrow a \times b > 0$ und

$$s_n(d_n(a) \times d_n(b)) = s_n((2^n + a) \times (2^n + b)) = s_n(2^{n+n} + 2^n \times a + 2^n \times b + a \times b)$$

$$= a \times b = d_n(a \times b)$$

Insgesamt folgt, dass die Behauptung (*) gilt.