Q. 長さnの配列の
ソートに必要な最小の比較回数は?

Q. 長さnの配列のソートに必要な最小の比較回数は?

A. 0回!!

Radix Sortを使えばよいため.

Radix Sortの例:

① 32, 42, 21, 11, 3

②

0	1	2	3	4	…
	21	32	3		
	11	42			

③

0	10	20	30	40	…
3	11	21	32	42	

Q. 長さnの配列の

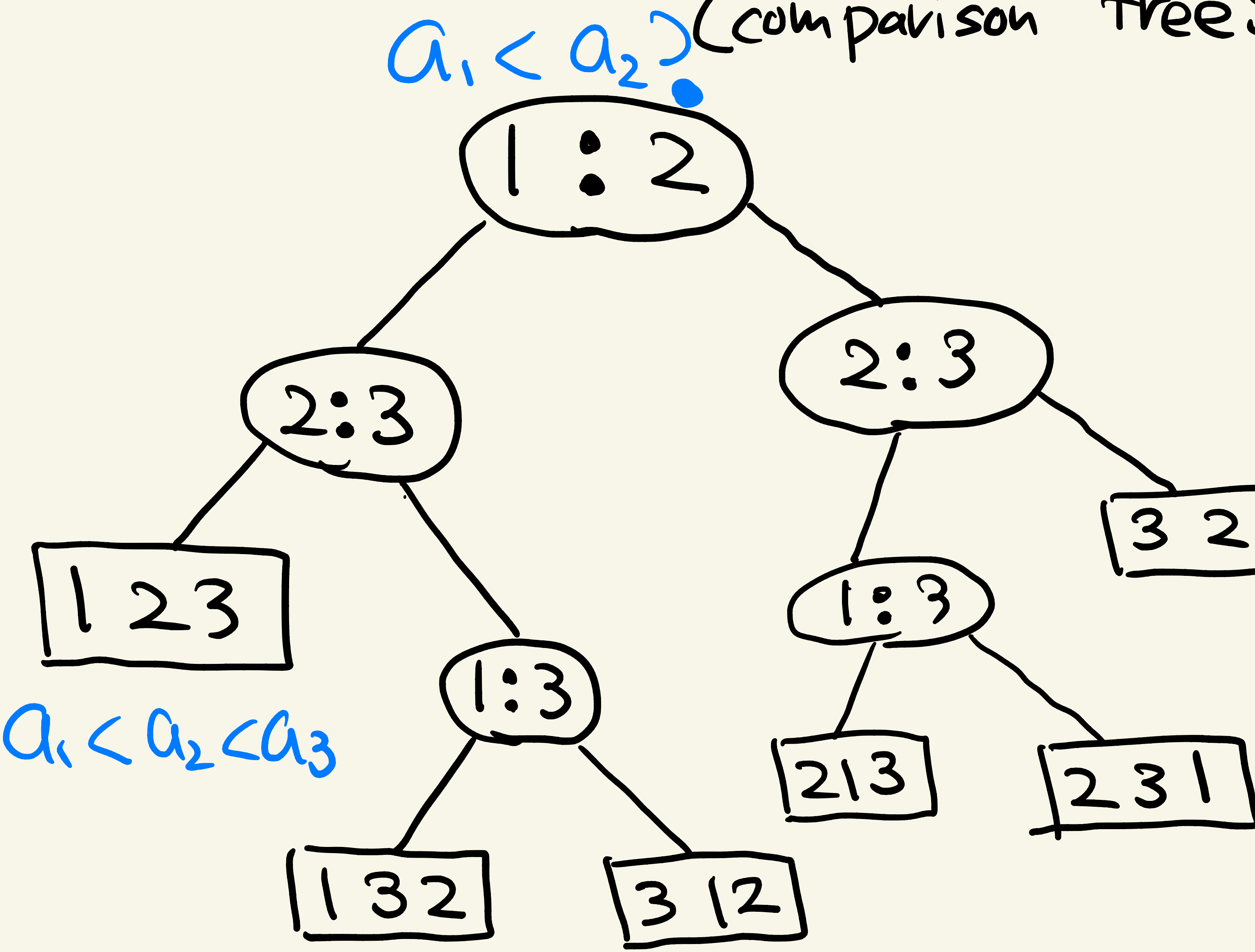
ソートに必要な最小の比較回数は?

「比較」ソートに限定する.

(あと, 配列の要素は互いに異なる自然数

とする)

比較ソートのモデル： 比較木 (comparison trees)



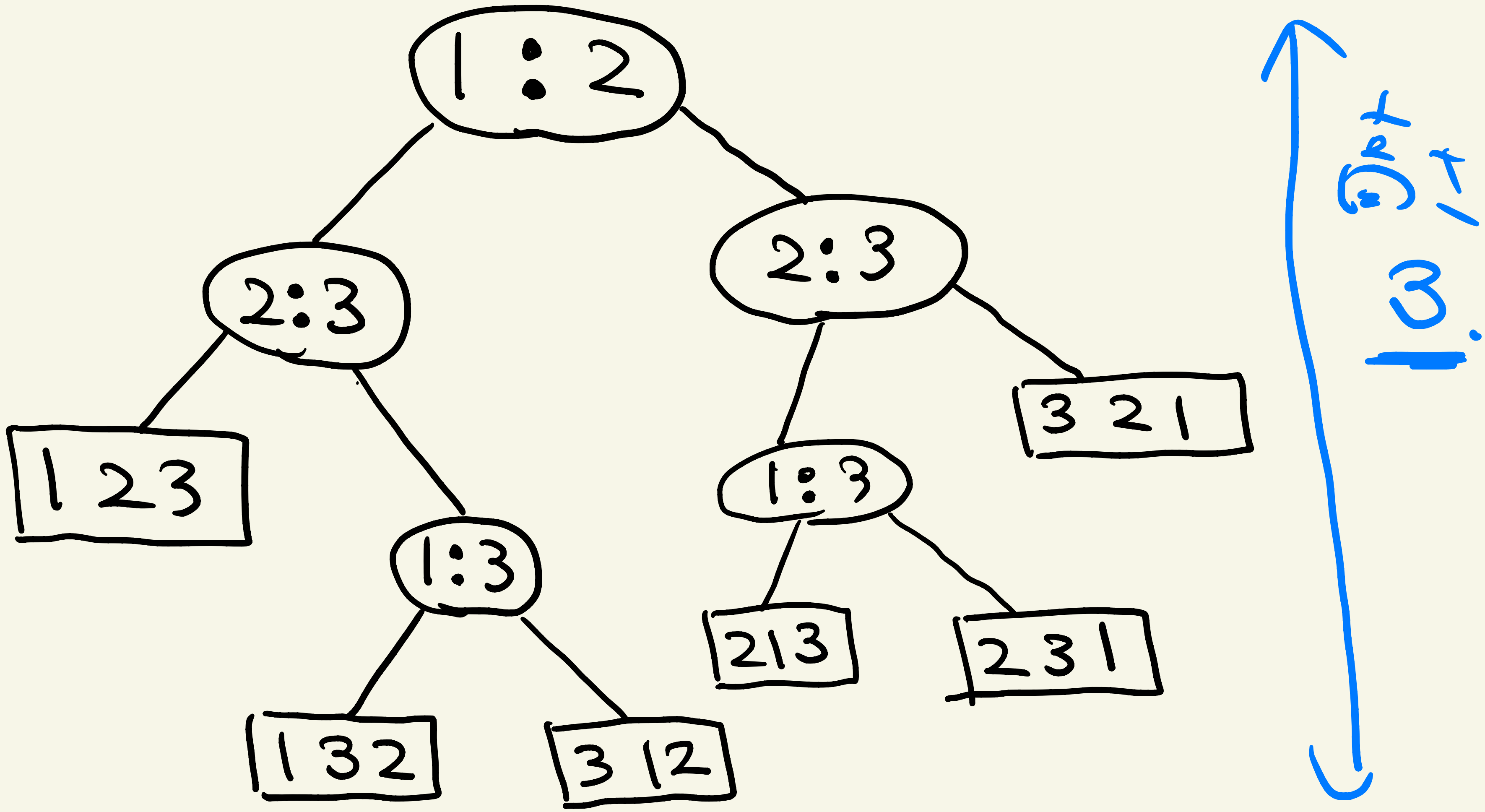
比較木とプログラムが対応づいている：

```
sort3(a : int[])
  if a[1] < a[2]
    if a[2] < a[3]
      return [a[1], a[2], a[3]]
    else
      if a[1] < a[3]
                ⋮
  else
          ⋮
```

最小の比較回数を求める問題は

高さ最小の比較木をつくる問題になった。

例:



比較木から「情報理論的下界」が得られる:

$C(n)$

葉の数が $n!$ 個の2分木の高さは

$\lceil \log_2(n!) \rceil$ 以上でなければならない.

$$\lceil \log_2(n!) \rceil = n\log_2(n) - n/\log_e(2) + \frac{1}{2}\log_2(n) + (\text{定数})$$

スターリングの公式: $n! \simeq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

$C(n) = \lceil \log_2(n!) \rceil$

$S(n)$: 比較ソートに必要な比較回数の最小値.

当然 $S(n) \geq C(n)$

n	1	2	3	4	5	6	...
$C(n)$	0	1	3	5	7	10	...
$S(n)$	0	1	3	5	7	10	

n=4のときの最適解: 5回

⓪ $a - b$

$c - d$

① $a' < b'$

$\quad\quad\;\; |$

$c' < d'$

② $a'' < b''$

$\quad\quad\;\; \wedge$

$c'' < d''$

③ $b'' - c''$

(④) $a'' - c''$

n=5 のときの最適解: 7回

⓪ n=4 のときと同じことをすると 3回 で以下がつくれる:

$$\begin{array}{ccccc} a & < & b & & \\ & & \wedge & & \\ c & < & d & & e \end{array}$$

eを先に入れるのがミソ

① 高々 2回 で $a<b<d$ のどこかに e を入れる:

$$\begin{array}{c|c|c|c} \begin{array}{c} e<a<b<d \\ \hfill\vee \\ \hfill c \end{array} & \begin{array}{c} a<e<b<d \\ \hfill\vee \\ \hfill c \end{array} & \begin{array}{c} a<b<e<d \\ \hfill\vee \\ \hfill c \end{array} & \begin{array}{c} a<b<d<e \\ \vee \\ c \end{array} \end{array}$$

② 高々2回で c を挿入。

この考察を一般化し、再帰的に行うのが

merge insertion アルゴリズム

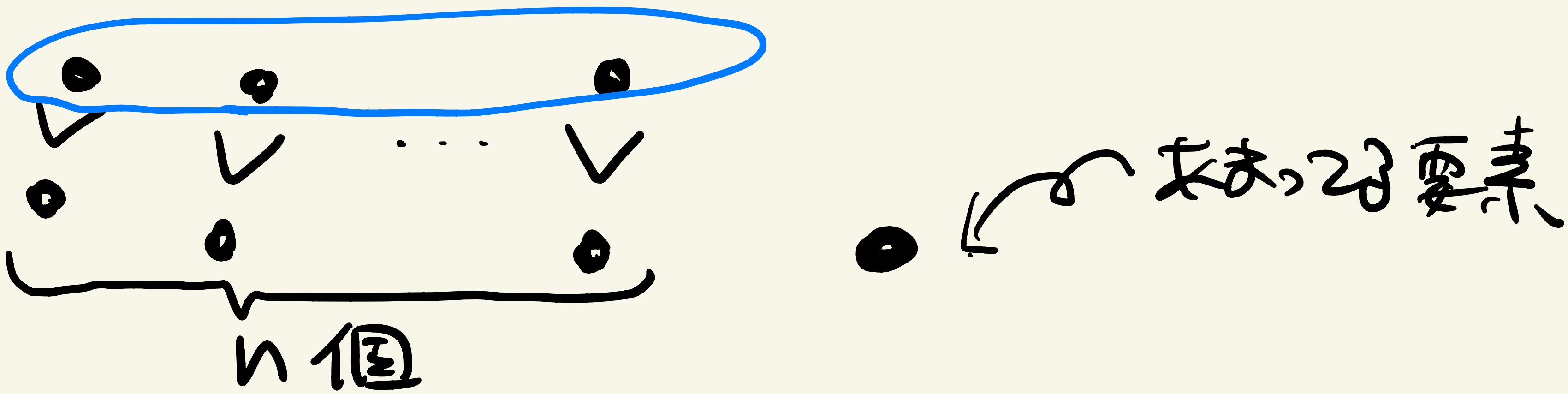
(Ford-Johnson アルゴリズム)

$F(n)$: n 要素に対し上のアルゴリズムを使ったときの比較回数.

当然 $C(n) \leq S(n) \leq F(n)$

merge insertion の sketch.

① $\lfloor n/2 \rfloor$ のペアの比較



② に merge insertion. 主鎖 (main chain)

$$a_1 < a_2 < a_3 < \cdots < a_n$$

$$b_1 \quad b_2 \quad b_3 \quad \cdots \quad b_n \quad b_{n+1}$$

③ $b_2, \dots, b_n, b_{n+1}$ を に「うまい順で」挿入.

n=7のとき

② まで行うと

主鎖

$$a_1 < a_2 < a_3$$

$$b_1 \qquad b_2 \qquad b_3 \qquad b_4$$

③ b_2, b_3, b_4 を に「うまい順で」挿入.

Q. どの順が最適？（正解は1つ）

- b_2, b_3, b_4
- b_2, b_4, b_3
- b_3, b_2, b_4
- b_3, b_4, b_2
- b_4, b_2, b_3
- b_4, b_3, b_2

正解は b_3, b_2, b_4.

一般の場合の「うまい順」は

$b_3, b_2; b_5, b_4; b_{11}, b_{10}, \cdots, b_6; \cdots;$

$b_{t_k}, b_{t_k-1}, \cdots, b_{t_{k-1}+1}; \cdots$

ここで $(t_1, t_2, t_3, t_4, \cdots) = (1, 3, 5, 11, \cdots)$

$$t_{k-1} + t_k = 2^k.$$

Hadian の Ph.D. thesis によると

$$F(n) = \sum_{k=1}^{n} \left\lceil \log_2\left(\frac{3}{4}k\right) \right\rceil$$

これで $F(n)$ を計算すると

$n = 1, \ldots, 11, 20, 21$ について

$F(n) = C(n)$ がわかる.

Q. merge insertionは最適か?

A. No!

- $n=47$について $S(n) < F(n)$ (Mönting 1981)
- 無限個の$n \in \mathbb{N}$について$S(n) < F(s)$ (Manacher 1979)

$C(n) < F(n)$ のときは最適な $S(n)$ をコンピュータで組みあわせ的に探索するアプローチが取られる.

(Wells や Peczarski の研究)

$n \leq 22$ については $n = 16, 17, 18$ の場合を除いて $S(n)$ が決定されたが、

2022年になり $n \leq 22$ について完全解決された.

(Stober and Weiss 2023)

まとめ

比較ソートの比較回数の最小値 $S(n)$

- 問題の定式化：比較木
- 情報理論的下限 $\lceil \log_2(n!) \rceil \sim n\log_2 n$
- merge insertion
- コンピューターによる $S(n)$ の探索の動向

参考：Knuth の AoCP vol3. Sec.5.3.1