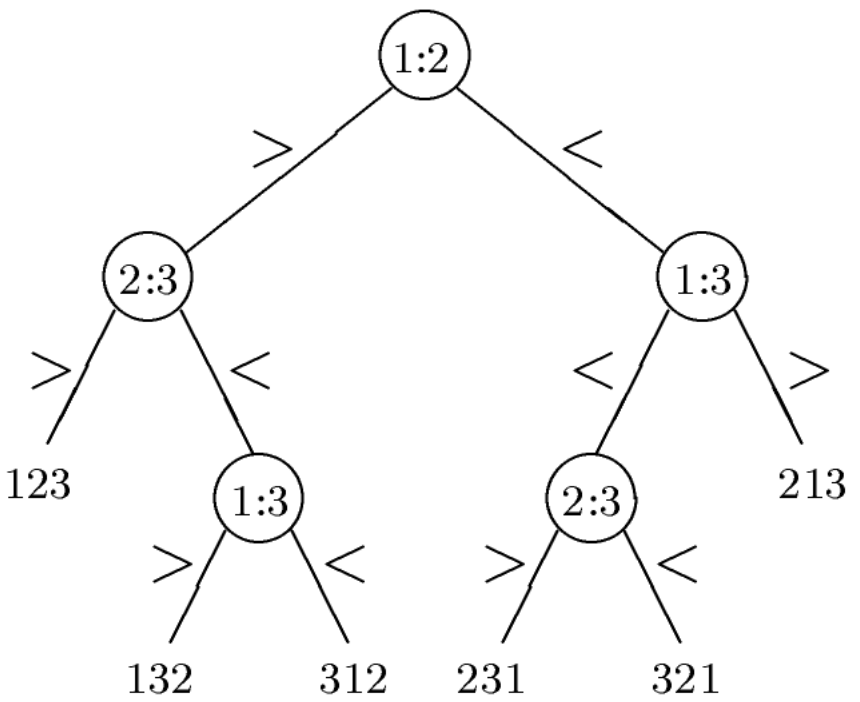
Asymptotically fast sorting

Sergeev I. S.

MVK seminar, 2021



$S(n)$ – minimal number of comparisons to sort n elements

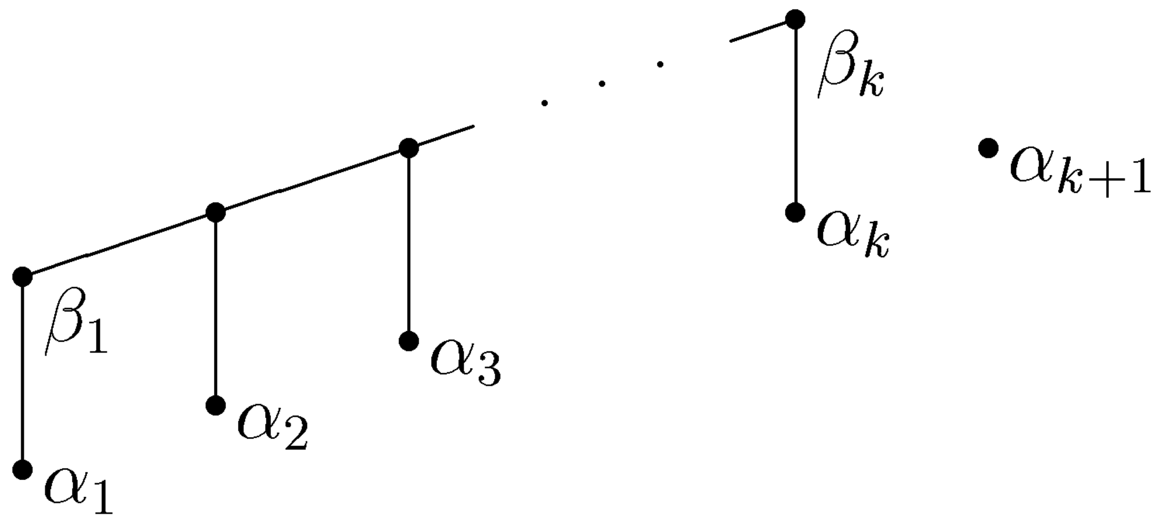
$$S(n) \geq \log_2(n!) \sim n \log_2 n$$

$$S(n) \leq \log_2(n!) + O(n) \sim n \log_2 n$$

1) Sorting by binary insertions
2) Sorting by trees
3) Sorting by mergings

Method of binary insertions

(S. Ford, L. Johnson'59)



$$S(n) \leq \log_2(n!) + c\,n + O(\log n)$$

$$c = \log_2(3e/8) \approx 0.028 \ (\text{for } n \sim 2^k/3) \ \ldots$$
$$\ldots \log_2(3/(4\ln 2)) \approx 0.114 \ (\text{for } n \sim \ln 2 \cdot 2^k/3).$$

$c < 0.07$ (G. Manacher, T. Bui, T. Mai'89)

Average complexity over all permutations of inputs:

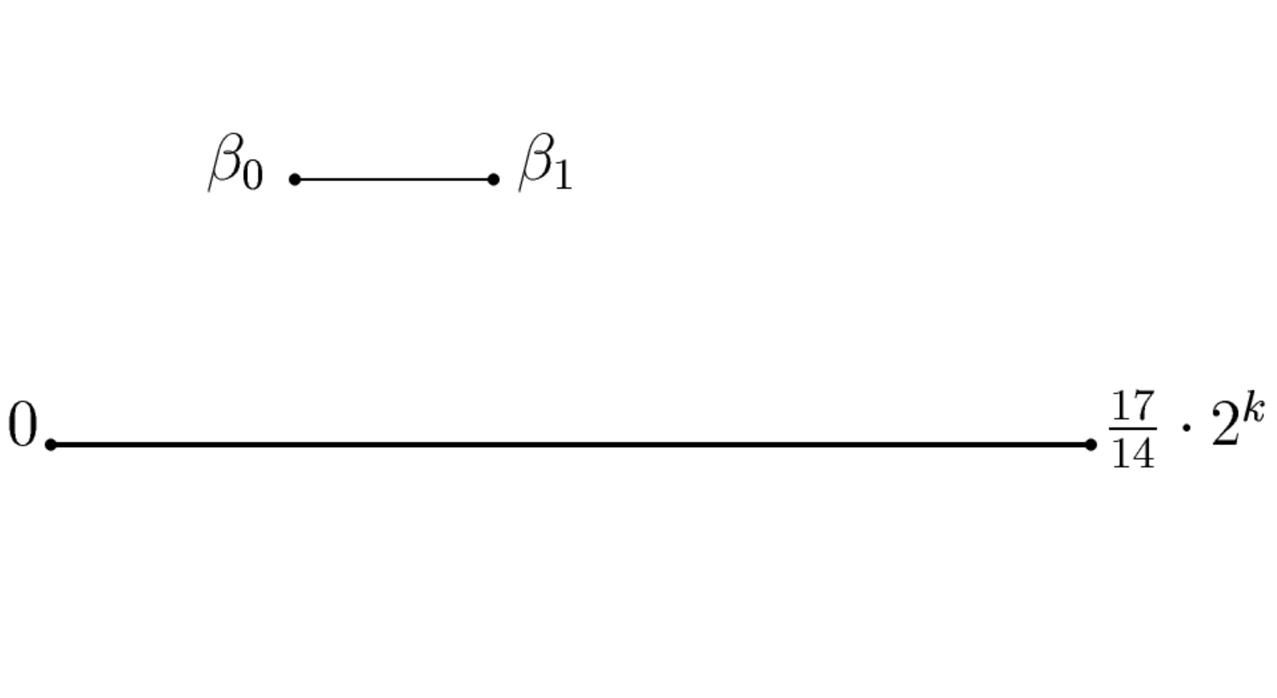
$c < 0.032$ (K. Iwama, J. Teruyama, S. Edelkamp, A. Weiß, S. Wild'20)

Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M(\tfrac{17}{14}\cdot 2^k - 1) = 2k, \quad M(\tfrac{6}{7}\cdot 2^k - 1) = 2k - 1$$

(R. Graham, F. Hwang, S. Lin'71)

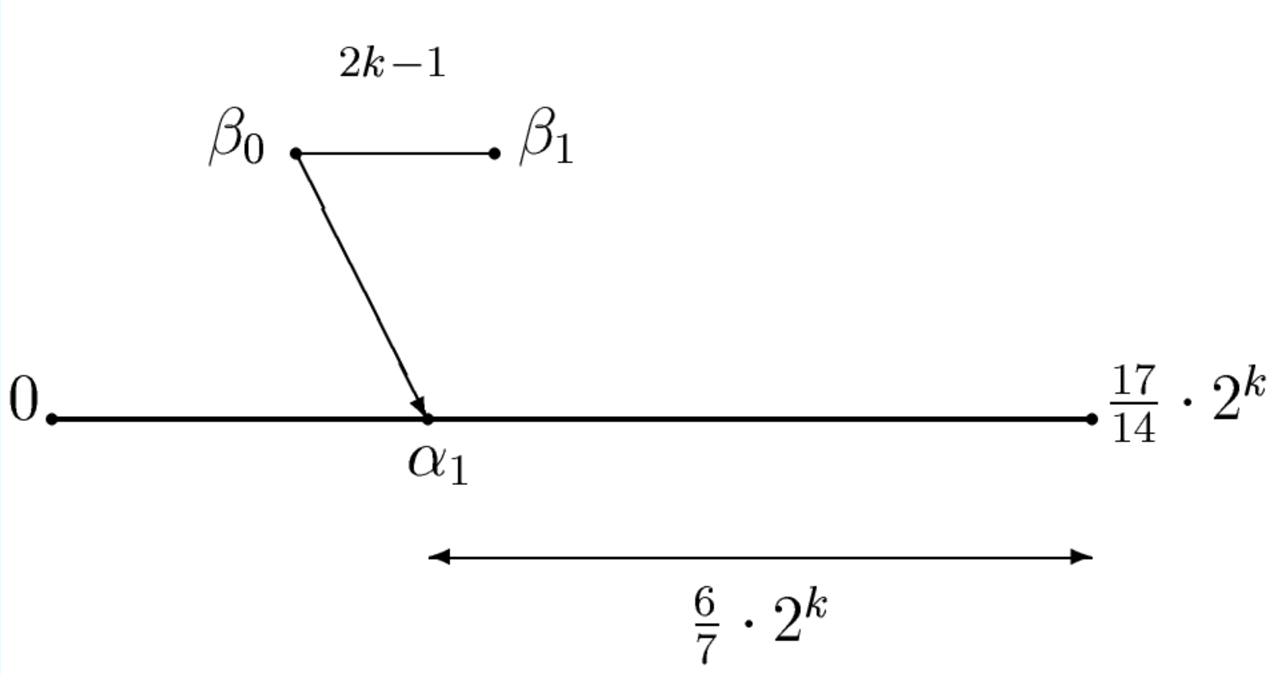


Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M(\tfrac{17}{14}\cdot 2^k - 1) = 2k, \quad M(\tfrac{6}{7}\cdot 2^k - 1) = 2k - 1$$

(R. Graham, F. Hwang, S. Lin'71)

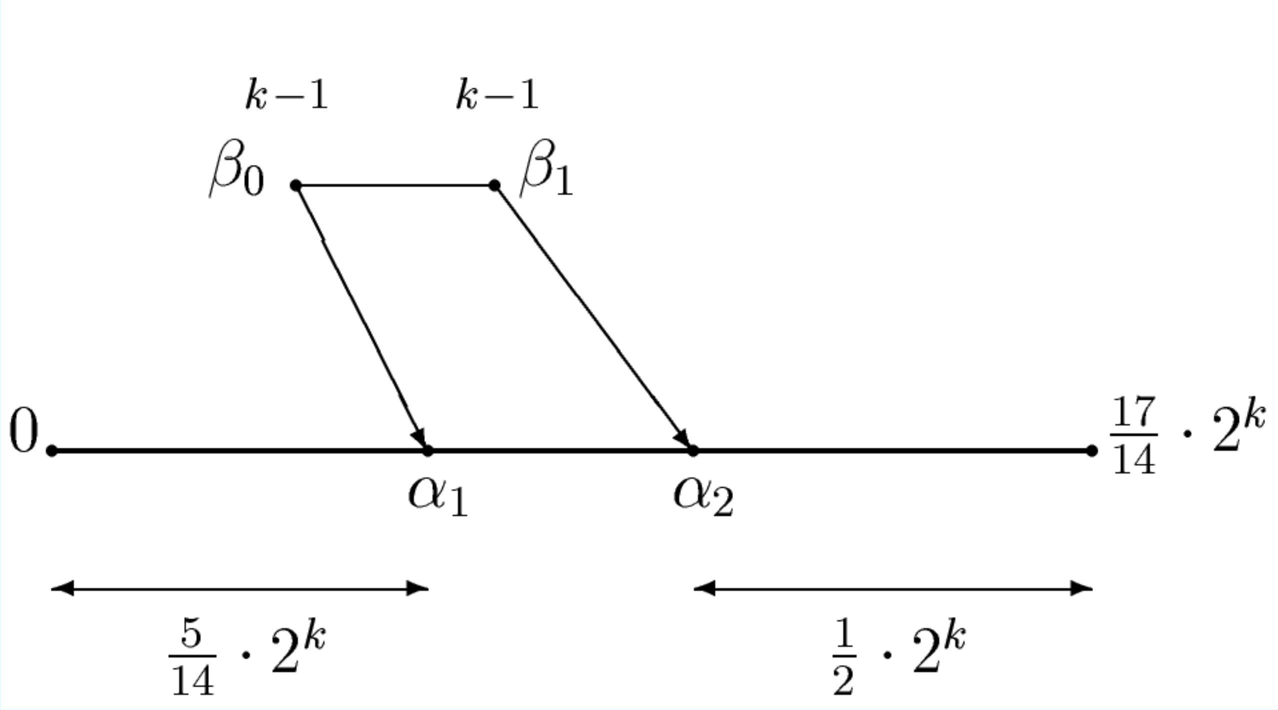


Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M(\tfrac{17}{14} \cdot 2^k - 1) = 2k, \quad M(\tfrac{6}{7} \cdot 2^k - 1) = 2k - 1$$

(R. Graham, F. Hwang, S. Lin'71)

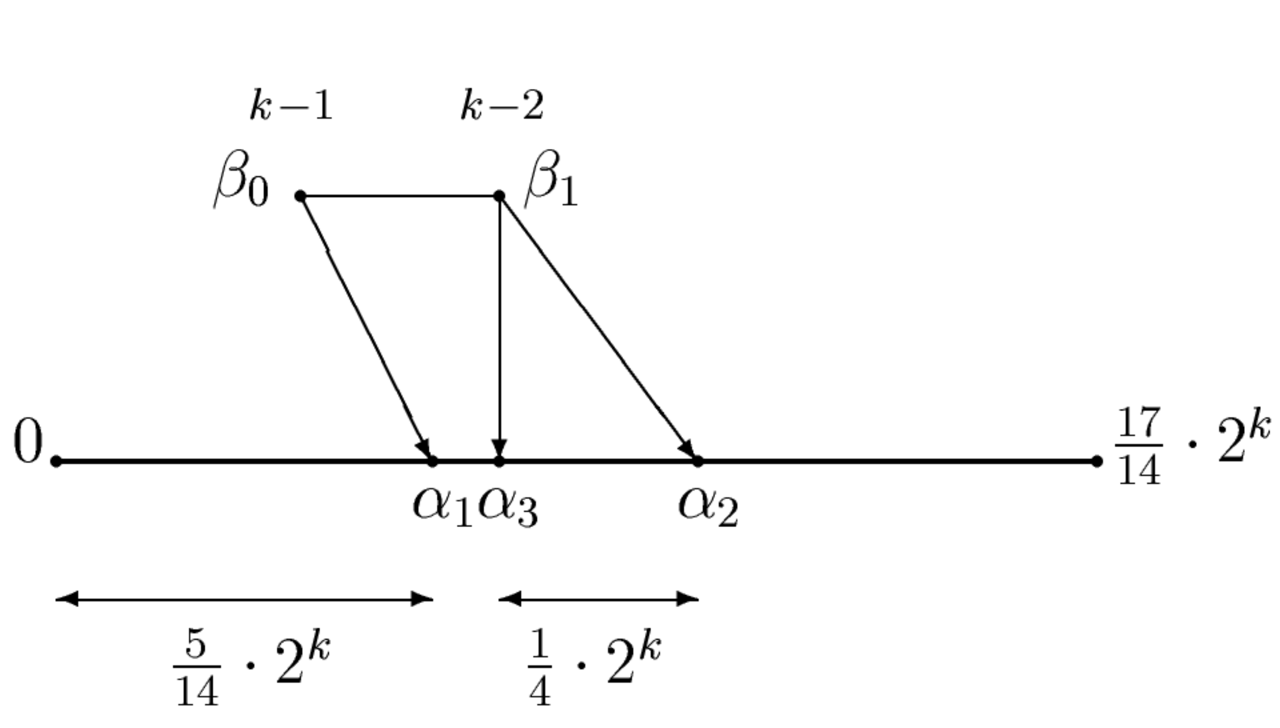


Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M\left(\tfrac{17}{14}\cdot 2^k - 1\right) = 2k, \quad M\left(\tfrac{6}{7}\cdot 2^k - 1\right) = 2k - 1$$

(R. Graham, F. Hwang, S. Lin'71)

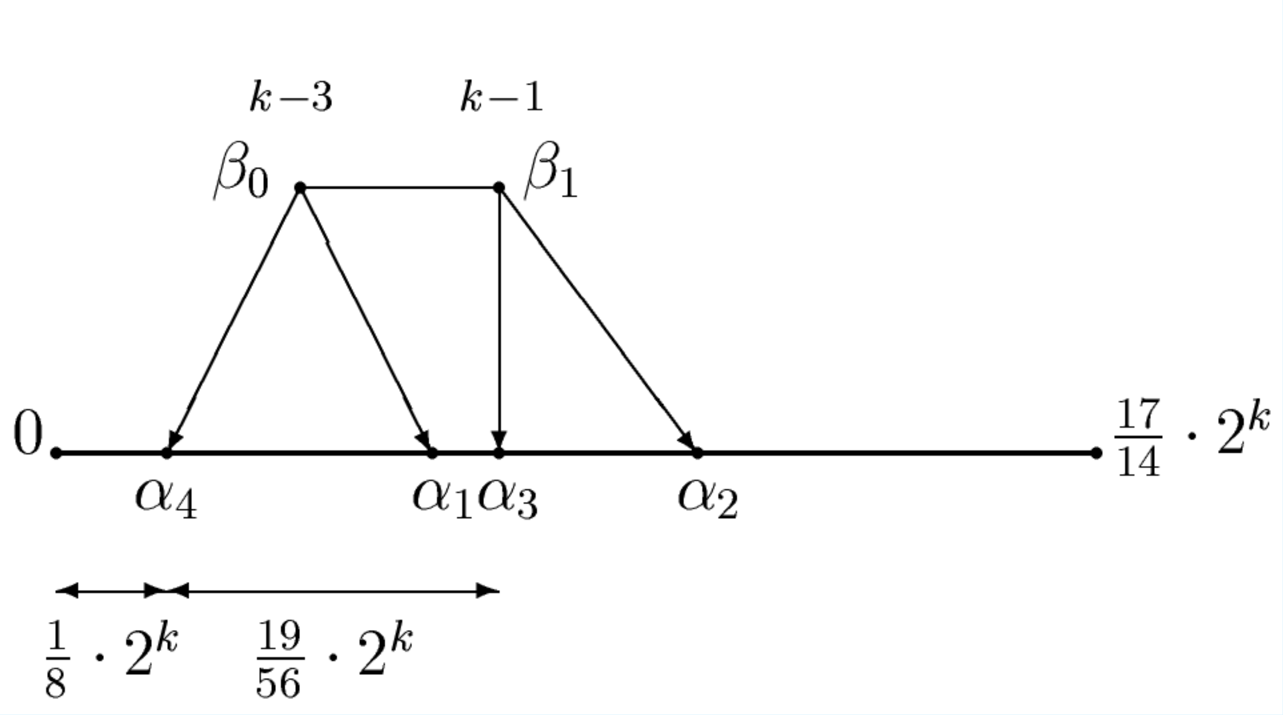


Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M(\tfrac{17}{14}\cdot 2^k-1)=2k,\quad M(\tfrac{6}{7}\cdot 2^k-1)=2k-1$$

(R. Graham, F. Hwang, S. Lin'71)

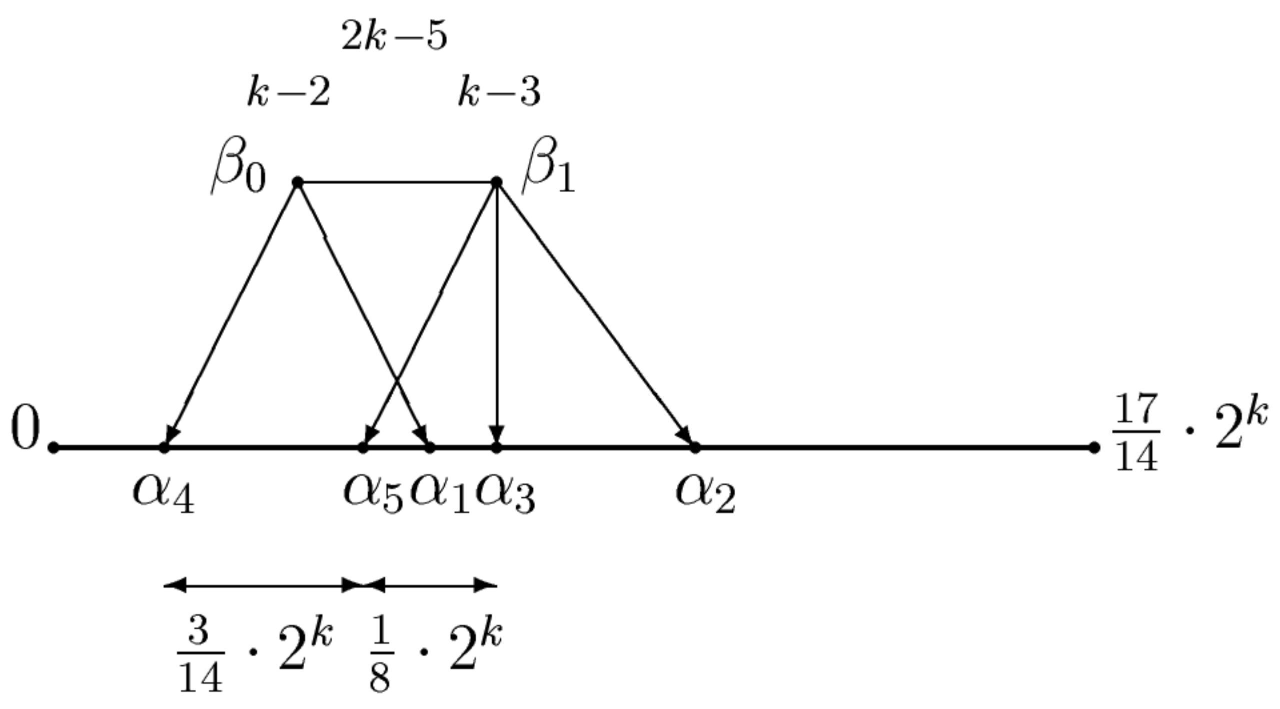


Group insertions

$M(n)$ – complexity of insertion of an ordered pair to a linearly ordered array of size n

$$M\left(\tfrac{17}{14}\cdot 2^k - 1\right) = 2k, \quad M\left(\tfrac{6}{7}\cdot 2^k - 1\right) = 2k - 1$$

(R. Graham, F. Hwang, S. Lin'71)

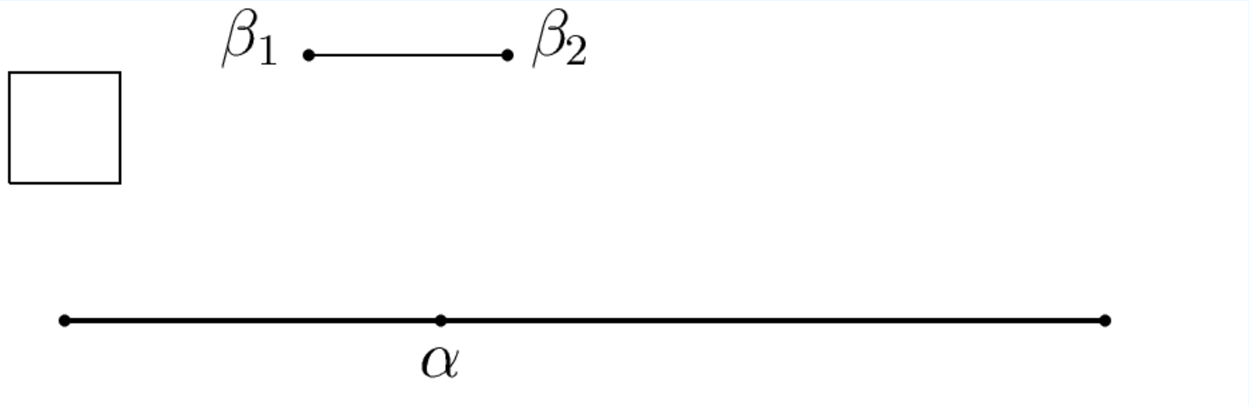


Goal: average complexity of the insertion of an element into an array of size n: $\log_2 n + \Theta(1) \to \log_2 n + o(1)$

Method of group insertions

Principles:

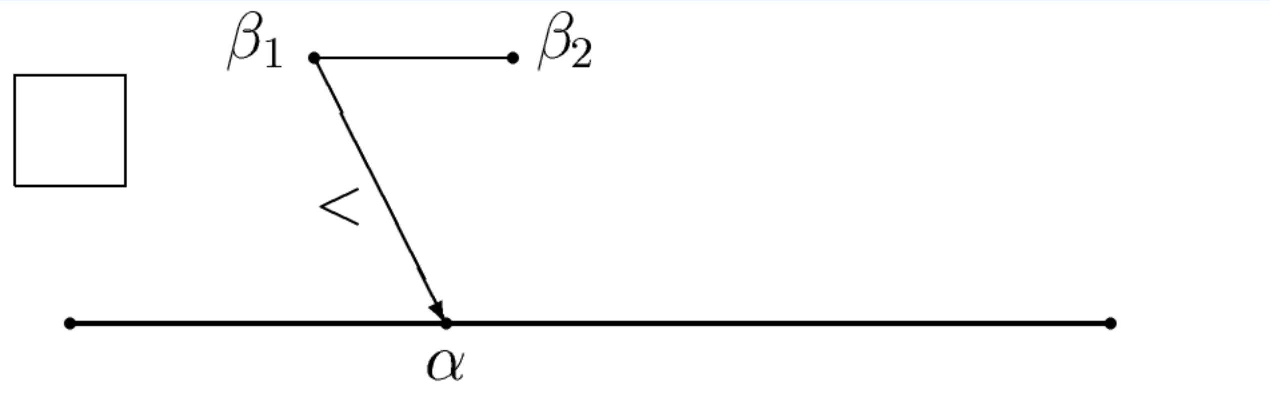
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

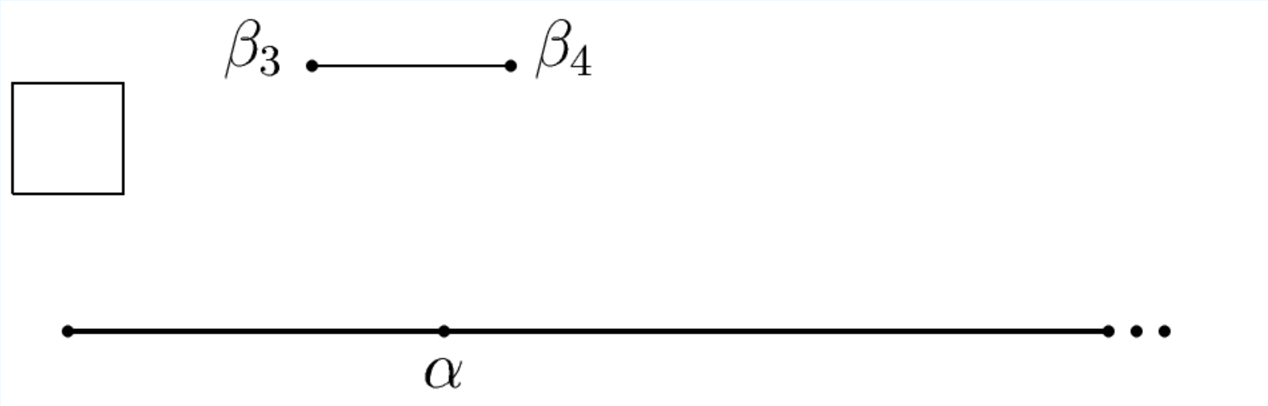
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

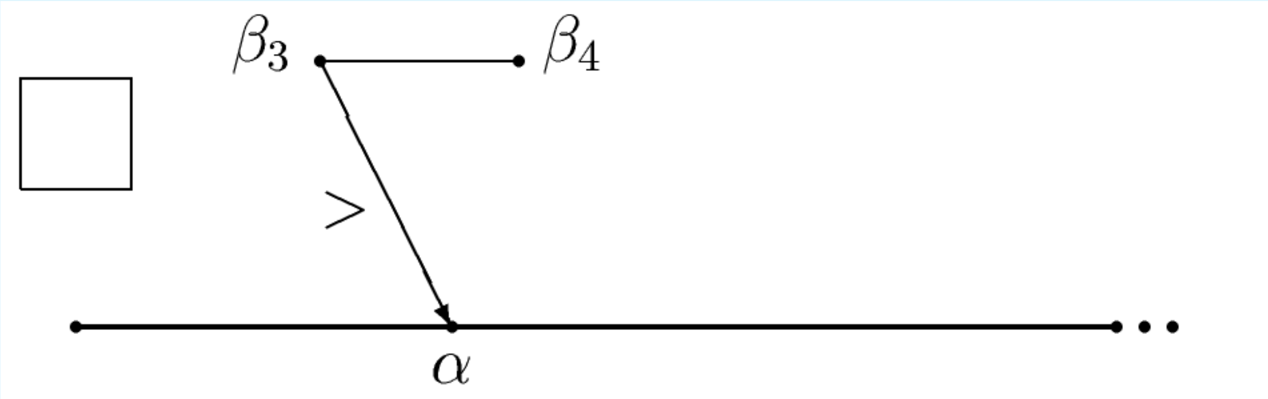
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

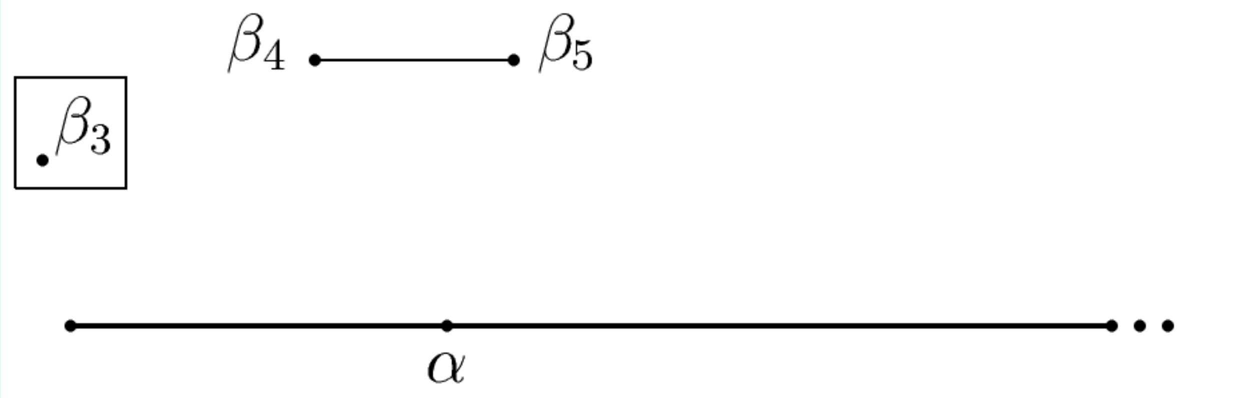
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

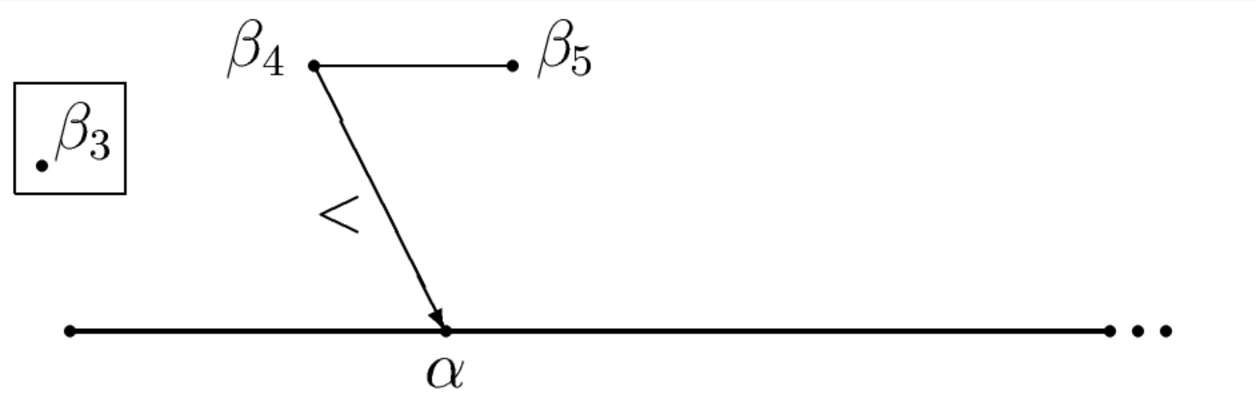
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

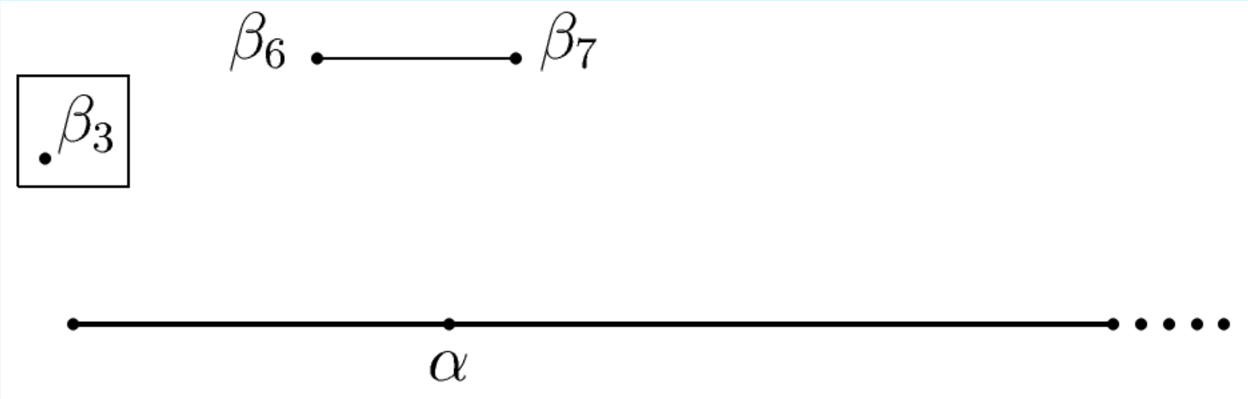
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

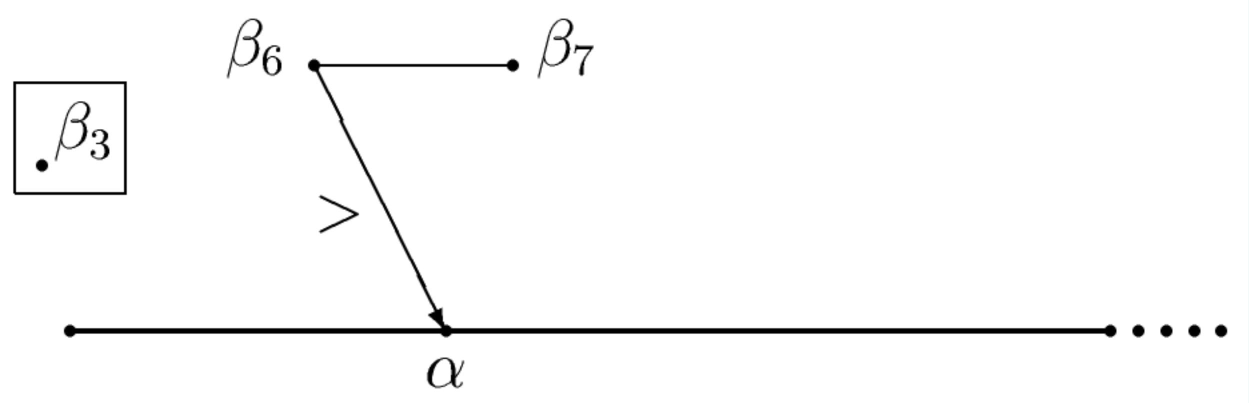
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



Method of group insertions

Principles:

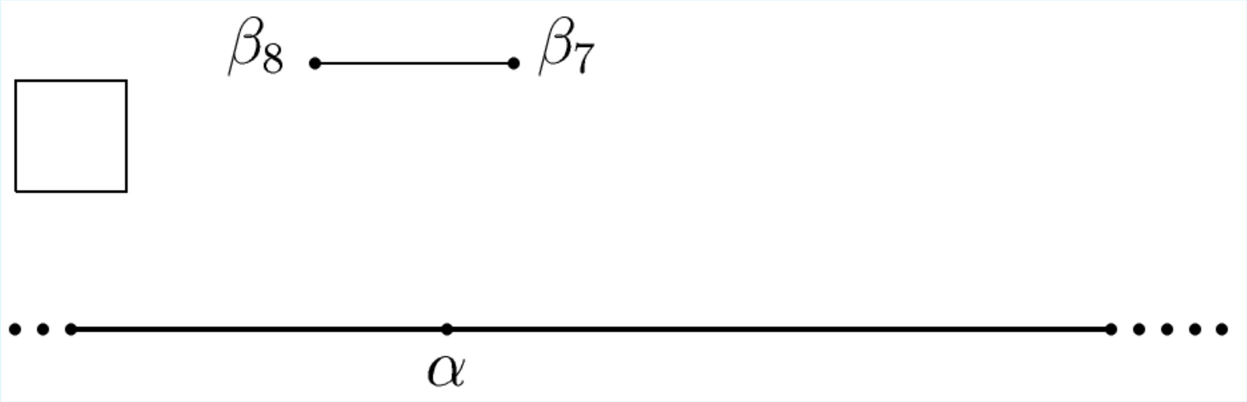
1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



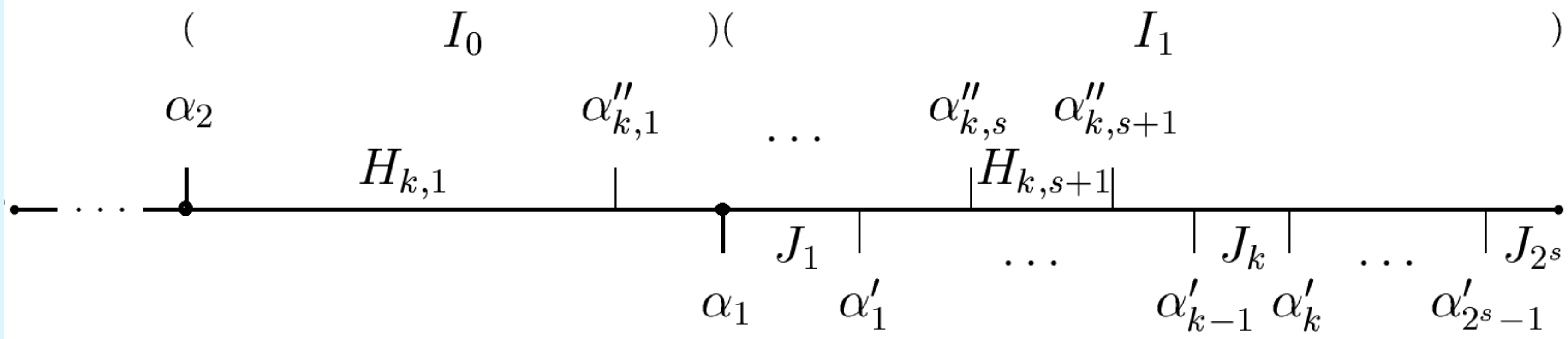
Method of group insertions

Principles:

1. A comparison divides the set of outcomes ≈ in half
2. Processing of ordered pairs
3. Containers to store single elements



example of partitioning an array in the method:



Method of group insertions (2)

T. insertion of $m \le 2^{a/2}$ ordered pairs into an ordered array of size $2^{a+1/2} - O(a^{-1/5} \cdot 2^a)$ costs at most $2am$ comparisons

complexity of insertion into array of size n per element: $\log_2 n + o(1)$

$$\Longrightarrow \qquad S(n) = \log_2(n!) + o(n)$$

1. Elements are divided into ordered pairs.
2. The larger elements of the pairs are sorted.
3. Insertion of the first $n/\log n$ lower elements.
4. The remaining lower elements are divided into groups of size $n^{1/2}/\log n$.
5. Groups are inserted one by one.

<u>PS</u>:

T(n) – complexity of selecting the median of n elements:

$(2+\varepsilon)n < T(n) < 2.95n$ (D. Dor, U. Zwick'95)

References:

1. Knuth D.E. *The art of computer programming*. *Vol.* 3. *Sorting and searching*. Reading: Addison-Wesley, 1998.
2. Sergeev I.S. *On the upper bound of the complexity of sorting*. Computational Mathematics and Mathematical Physics. 2021, 61(2), 329–346.