1 3-Wise Comparisons

Assume you have a function called 3-sort that takes as input an array of size 3 of comparable objects and returns them in sorted order. You want to sort an array of size \( n \geq 3 \) objects only using calls to 3-sort. How many function calls of 3-sort are required to sort an array of size \( n \)? Choose the asymptotically largest applicable lower bound.

- \( \Omega(\log n) \)
- \( \Omega(\sqrt{n}) \)
- \( \Omega(n) \)
- \( \Omega(n \log n) \)

Correct

2 Possible or Impossible

Is it possible to implement the following algorithms or data structures?

A data structure that stores comparable objects and supports the following operations:

- Insert new objects in \( O(1) \) time per insertion.
- Remove objects in \( O(1) \) time per removal.
- Return the smallest object currently in the collection in \( O(1) \) time.

Is it possible to implement such a data structure?

- Possible
- Impossible

Correct

An algorithm that runs in \( O(n) \) time and takes an array \( A \) of size \( n \) of comparable objects, \( i \), and \( j \) (\( 1 \leq i < j \leq n \)) as inputs and returns all the elements that are greater than the \( i \)th smallest element in \( A \) and less than the \( j \)th smallest element in \( A \)?

- Possible
- Impossible

Correct

Which algorithm would be a good choice to use for the last part?

- \( k \)-Select
- Radix Sort
- Quick Sort
- Merge Sort

Correct

An algorithm that runs in \( O(n) \) time and takes an array \( A \) of size \( n \) of comparable objects and partitions \( A \) into \( \frac{n}{\log n} \) groups of size \( \log n \) each, where for every \( i < j \), members of group \( i \) are smaller than or equal to members of group \( j \).

- Possible
- Impossible

Correct

3 Radix Sort and Counting Sort

Assume you have an array \( A \) of size \( n \) with positive integer element with all elements in the range \( [1, n^3] \).

What is the runtime of Counting Sort run on \( A \)?

- \( \Theta(n) \)
- \( \Theta(n \log n) \)
- \( \Theta(n^2) \)
- \( \Theta(n^3) \)

Correct

What is the runtime of Radix Sort, using base 10, run on \( A \)?

- \( \Theta(n) \)
- \( \Theta(n \log n) \)
- \( \Theta(n^2) \)
- \( \Theta(n^3) \)

Correct

Which base for Radix Sort will result in the fastest Radix Sort algorithm to sort \( A \) with?

- 2
- 10
- \( n \)
- \( n^2 \)

Correct