

Awaking drenched in sweat one night, you clearly see your path to fame and fortune. You will build a robotic rhinoceros and tour the country singing songs about nature to children, who will be allowed to play and interact with the rhinoceros. While a real rhinoceros would be too dangerous, you believe a robotic rhinoceros can be kept in check. In each of the situations below, which sort would you use? In all cases, assume memory is not an issue, and that the goal is to minimize run time so that the rhino can react as quickly as possible to any potential trouble. Answers may be used many times.

----- The rhinoceros is outfitted with a large number of sensors, each of which generates objects of type `Observation`. Observations include many instance variables, including `importance`, `timestamp`, `pressure`, `temperature`, `light intensity`, etc. These are placed in an unsorted array, and every time 1000000 `Observations` are generated, they are delivered to a central processing unit that sorts the `Observations` by the `importance` field, which is of type `double`. What sort should you use to minimize the run time required to sort all `Observations` by `importance`?

- A. Quicksort
- B. Mergesort
- C. Insertion sort
- D. Selection sort
- E. Knuth shuffle

----- Due to some close calls, you're going to refactor the sorting process to deal with a rare but dangerous situation where some `Observations` are generated with an incorrect `importance` value. For engineering reasons not described here, you can detect these by sorting by the `timestamp` and `importance` of each `Observation`.

Instead of `importance`, you first want to sort by the `timestamp` of each `Observation`. The `timestamp` is of a comparable type called `DateTime`. What sort should you use to minimize the run time required to sort all 1000000 `Observations` by `timestamp`?

----- After sorting by `timestamp`, you want to sort by `importance` such that all the objects of the same `timestamp` stay clustered. What sort should you use to minimize the run time while maintaining this clustering?

----- You iterate through the array, update the `importance` of the very rare bad `Observations` with a new value, and sort once more. What sort do you use to put items in order of `importance`?

while minimizing run time?

Permutations. Given two arrays of N Point2D objects, design a subquadratic algorithm to count the number of points that appear more than once.

Inversions. Design a subquadratic algorithm that counts the number of inversions in an array.

Decimal dominants. Design an expected linear time algorithm to find all values that occur more than $N/10$ times in an array of size N .