

stable sorting

A sorting algorithm is *stable* if for any two entries x and y of S such that

- $x.getKey() == y.getKey()$ and
- x precedes y in S before sorting
- x precedes y after S is sorted

sorting



bucket sort
aka
pigeonhole sort

sorting



PIGEONHOLE SORT

bucket sort
aka
pigeonhole sort



bucket sort (aka pigeonhole sort)

sorting

Use the key of an object as an index into a bucket array

Assume we have n object to sort

Assume keys are in range $[0, N-1]$



insert object e into `bucket[e.getKey()]`

Complexity is $O(n+N)$

bucket sort (aka pigeonhole sort)

sorting

Use the key of an object as an index into a bucket array

Assume we have n object to sort

Assume keys are in range $[0, N-1]$



insert object e into `bucket[e.getKey()]`

Complexity is $O(n+N)$

How might we do this in java?

What data structures would we use?

Is it stable?

What kind of data sets would be suitable?

SORTED