【DS】Day11

Tags	
Date	@June 4, 2022
Summary	3-way Partitioning

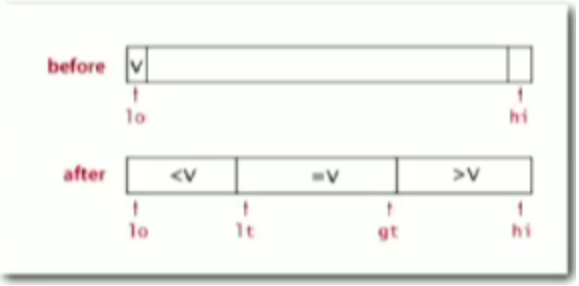
【Week3】Quicksort

3.6 Duplicate Keys

3-way partitioning

Goal: partition array into 3 parts so that:

- Entries between `lt` and `gt` equal to partition item v
- No larger entries to left of lt
- No smaller entries to right of gt



- Let v be partitioning item a[lo]
- Scan i from left to right
  - (`a[i] < v`): exchange a[lt] with a[i]; increment both lt and i
  - (`a[i] > v`): exchange a[lt] with a[i]; decrement gt
  - (`a[i] == v`): increment i

```
private static void sort(Comparable[] arr, int lo, int hi) {
  if (hi <= lo) return;
```

```
  int lt = lo, gt = hi;
  int i = lt;

  while (i <= gt) {
    int cmp = a[i].compareTo(v);
    if (cmp < 0)exch(arr, lt++, i++);
    else if (cmp > 0) exch(arr, i, gt--);
    else i++;
  }

  sort(arr, lo, lt - 1);
  sort(arr, gt + 1, hi);
}
```