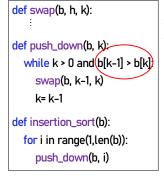
18

## **Algorithm Complexity (Q)**



Count (approximately) the number of comparisons needed to sort a list of length n

- A.  $\sim 1$  comparison B.  $\sim$  n comparisons
- C.  $\sim n^2$  comparisons

\_\_\_\_\_

17

19

## **Algorithm Complexity (A)**

- Count the number of comparisons needed
- In the worst case, need i comparisons to push down an element in a sorted segment with i elements.
- For a list of length n
  - 1<sup>st</sup> push down: 1 comparison
  - 2<sup>nd</sup> push down: 2 comparisons (worst case) :
  - $1+2+...+(n-1) = n^{*}(n-1)/2$ , say,  $n^{2}$  for big n
- For fun, check out this visualization: <u>https://www.youtube.com/watch?v=xxcpvCGrCBc</u>

## Complexity of algorithms discussed so far

- Linear search: on the order of n
- Binary search: on the order of log<sub>2</sub> n
  - Binary search is faster but requires sorted data
- Insertion sort: on the order of n<sup>2</sup>
- Next, let's look at merge sort