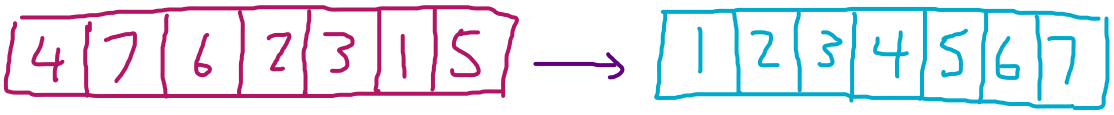
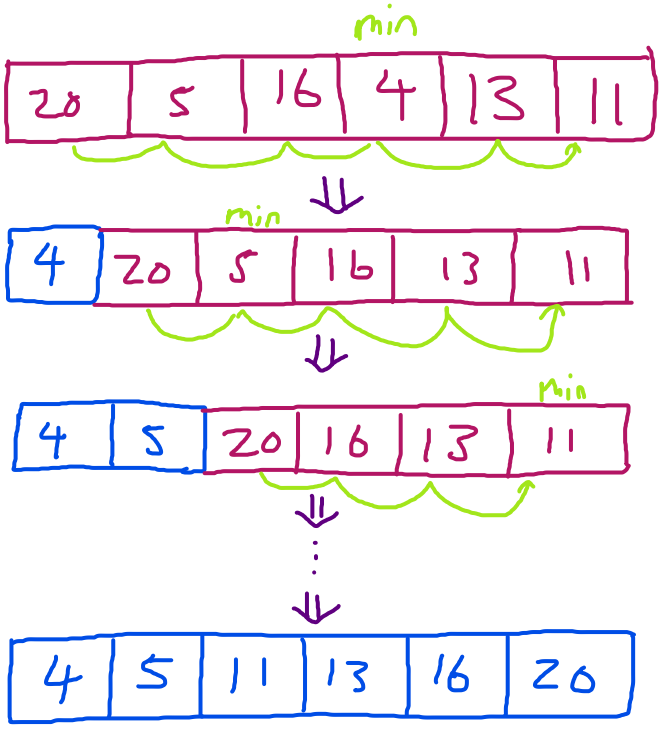
# Sorting




## Selection Sort

1. Find smallest item by traversing array
2. Swap this item to front of "unfixed" items and "fix" it
3. Repeat 1 and 2 for unfixed items until all items are "fixed"

Example:




## Selection Sort Runtime? $\Theta(N^2)$

We traverse the array to find the minimum. This will take $\Theta(N)$.
We repeat this $N$ times. Therfore, our runtime is $\Theta(N^2)$.

Stable? (equal items in unsorted array remains in same sequence)

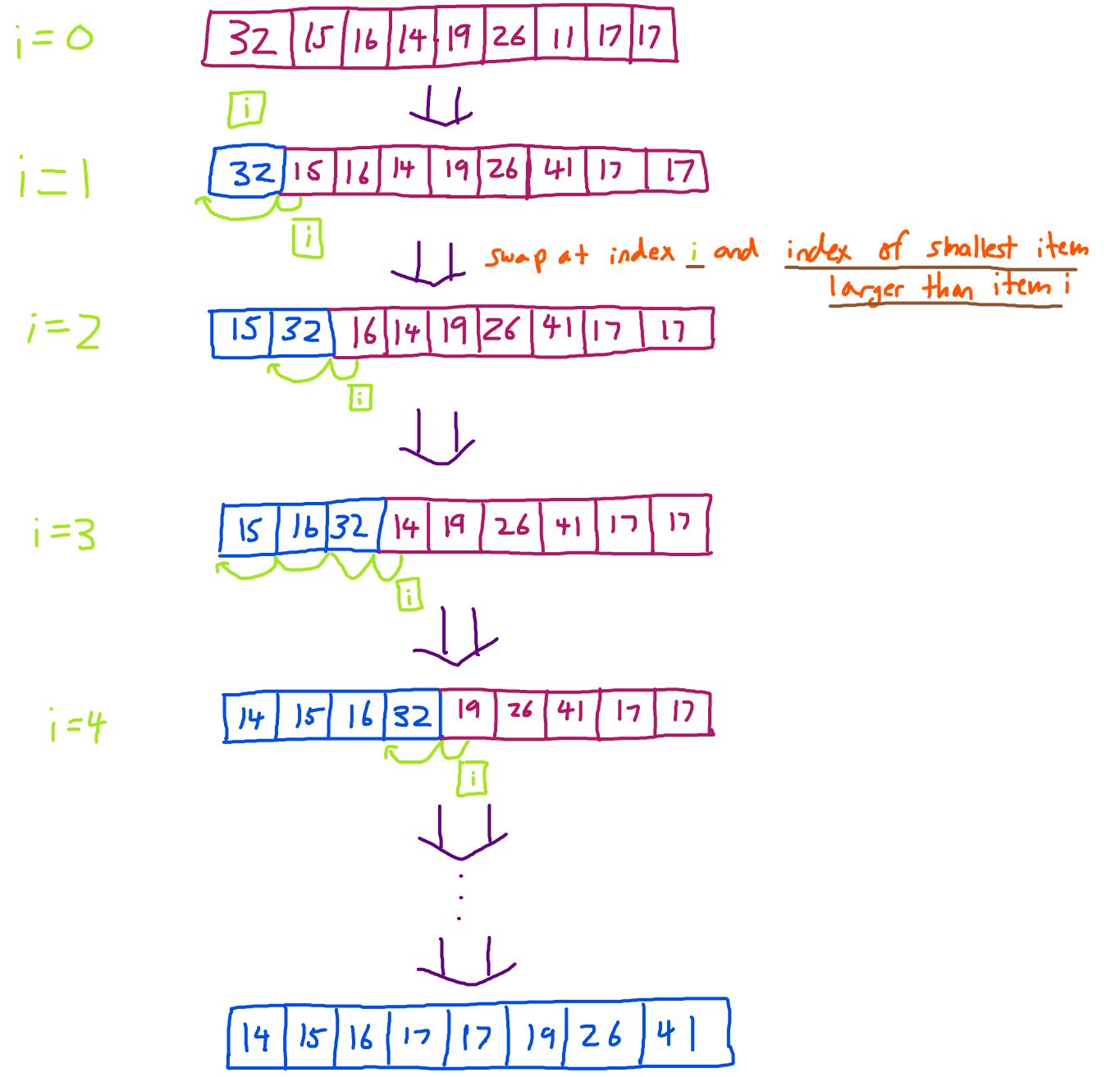
i.e. 

# Insertion Sort

Repeat for $i = 0, 1, \ldots, N-1$:

Swap item i backwards until it is in right place among previously examined items

Example:




- Runtime? $\Omega(N)$, $O(N^2)$

If array already well sorted, then no need for many swaps backwards

1 2 3 4 5 6

swap backwards 0 times, once we get to item i, we only look at prev item then proceed

- Stable?

Yes! When we try to swap item backwards, we stop trying when we see an item that is equal