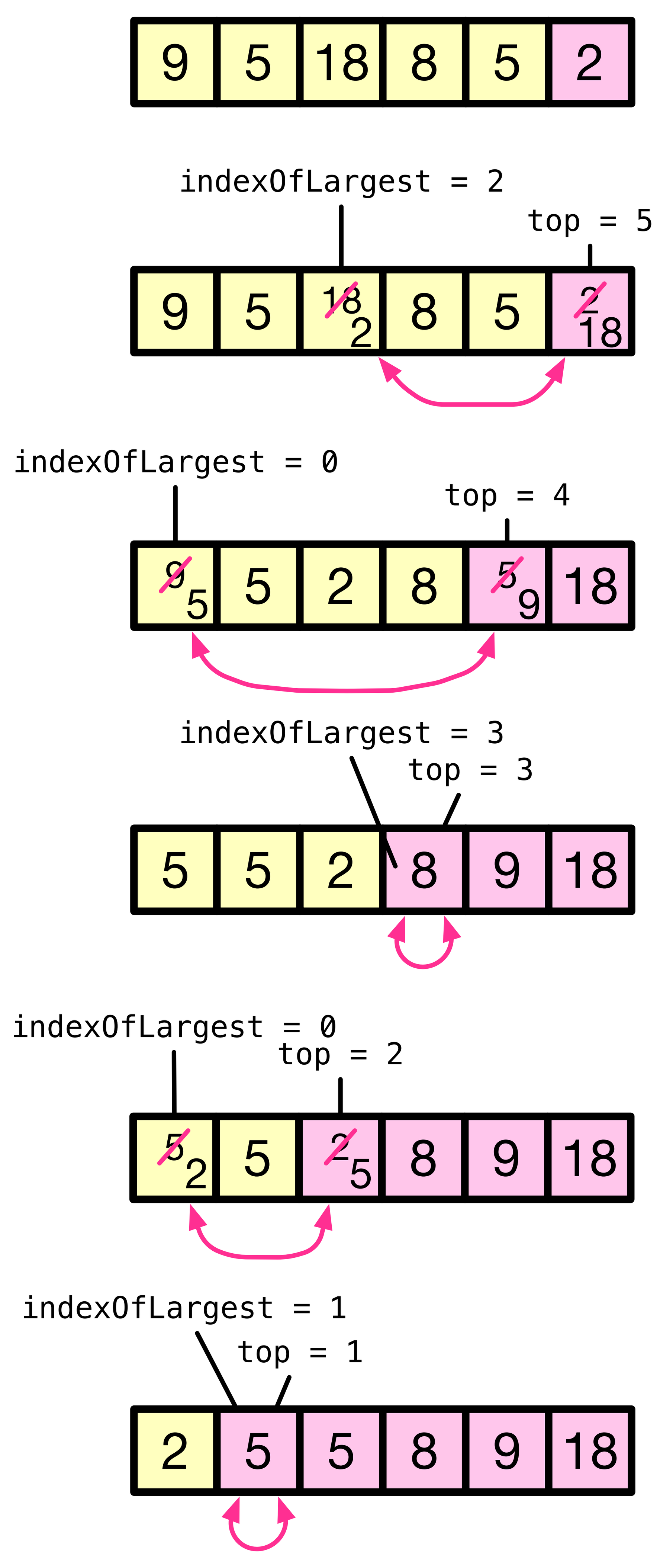
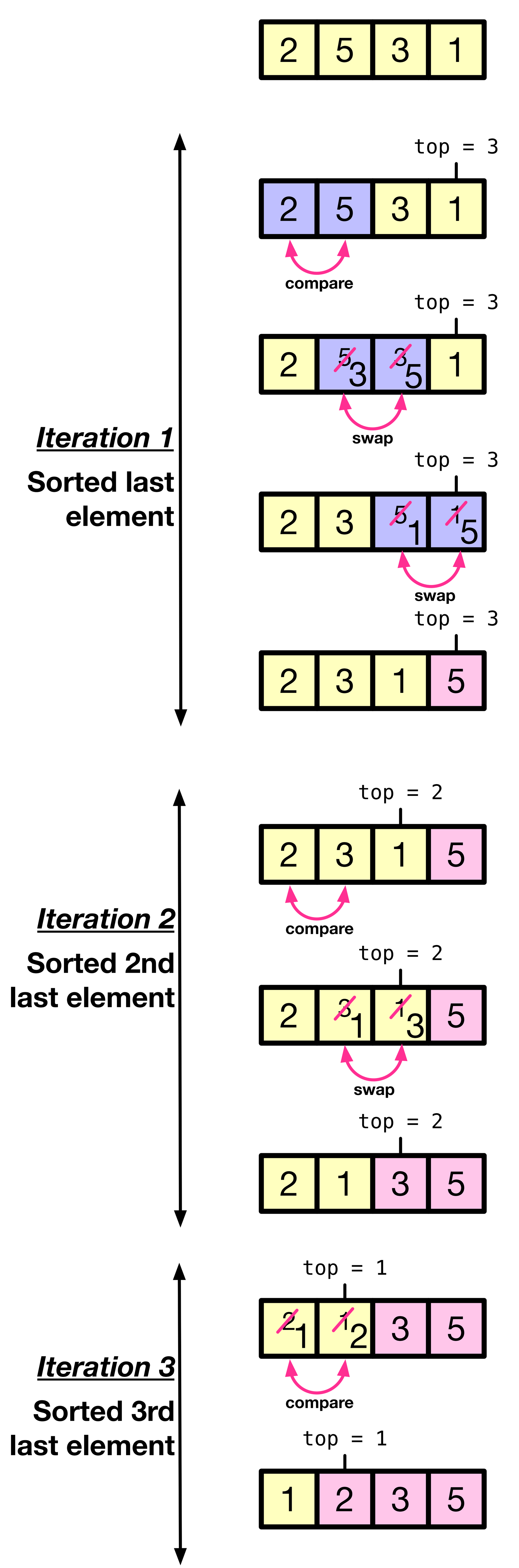

9 2 6 5 1 7
top = 1
compare A[top] with A[top − 1]
top = 1
insert A[top] in correct position A[0]
with respect to the first 2 elements
After first iteration, first two elements are sorted
top = 2
compare A[top] with A[top − 1], A[top − 2]
top = 2
insert A[top] in correct position A[top − 1]
with respect to the first 3 elements
After second iteration, first three elements are sorted
top = 3
compare A[top] with A[below top till 0]
top = 3
shift all elements that are > A[top] to right,
and insert A[top] in its correct position with
respect to the first 4 elements
After third iteration, first four elements are sorted
top = 4
compare A[top] with A[below top till 0]
top = 4
shift all elements that are > A[top] to right,
and insert A[top] in its correct position with
respect to the first 5 elements
After forth iteration, first four elements are sorted
top = 5
compare A[top] with A[i] till A[top] > A[i]
top = 5
shift all elements that are > A[top] to right,
and insert A[top] in its correct position with
respect to the first 6 elements
After fifth iteration, first six elements are sorted


9 5 18 8 5 2
indexOfLargest = 2
top = 5
indexOfLargest = 0
top = 4
indexOfLargest = 3
top = 3
indexOfLargest = 0
top = 2
indexOfLargest = 1
top = 1
2 5 5 8 9 18


2 5 3 1
top = 3
compare
swap
Iteration 1
Sorted last
element
top = 2
Iteration 2
Sorted 2nd
last element
top = 1
Iteration 3
Sorted 3rd
last element
1 2 3 5