Insertion Sort Example

A = 12,9,3,7,14,11

i = 0 , 1 , 2 , 3 , 4 , 5

for i= 1 to i=5:

set the key to A[i] (A[1]= 9). Hold the key to create an empty space.

9

A= 12, ,3,7,14,11

working with items at indices i-1 -> 0,

while the key is less than each item, move each up by 1 and then put the key in the empty space.

A= 9,12,3,7,14,11

increment i (now is i=2)

---

set the key to A[i] (A[2]= 3). Hold the key to create an empty space.

3

A= 9,12, ,7,14,11

working with items at indices i-1 -> 0,

while the key is less than each item, move each up by 1 and then put the key in the empty space.

A= 3,9,12,7,14,11

increment i (now is i=3)

---

set the key to A[i] (A[3]= 7). Hold the key to create an empty space.

7

A= 3,9,12, ,14,11

working with items at indices i-1 -> 0,

while the key is less than each item, move each up by 1 and then put the key in the empty space.

A= 3,7,9,12,14,11

increment i (now is i=4)

---

set the key to A[i] (A[4]= 14). Hold the key to create an empty space.

14

A= 3,9,7,12, ,11

working with items at indices i-1 -> 0,

while the key is less than each item, move each up by 1 and then put the key in the empty space.

A= 3,7,9,12,14,11

increment i (now is i=5)

---

set the key to A[i] (A[5]= 11). Hold the key to create an empty space.

11

A= 3,9,7,12,14,

working with items at indices i-1 -> 0,

while the key is less than each item, move each up by 1 and then put the key in the empty space.

A= 3,7,9,11,12,14

---

we have reached the end of the array and the sort is complete

Selection Sort Example

A = 12,9,3,7,14,11

i = 0 , 1 , 2 , 3 , 4 , 5

for i= 0 to i=4:

i=0

find the smallest item by checking all items between i=0 and i=5 (inclusive)
and swap the smallest item with the item at at i=0

A= 12,9,3,7,14,11

A= 3,9,12,7,14,11

increment i (now is i=1)

---

find the smallest item by checking all items between i=1 and i=5 (inclusive)
and swap the smallest item with the item at at i=1

A= 3,9,12,7,14,11

A= 3,7,12,9,14,11

increment i (now is i=2)

---

find the smallest item by checking all items between i=2 and i=5 (inclusive)
and swap the smallest item with the item at at i=2

A= 3,7,12,9,14,11

A= 3,7,9,12,14,11

increment i (now is i=3)

---

find the smallest item by checking all items between i=3 and i=5 (inclusive)
and swap the smallest item with the item at at i=3

A= 3,7,9,12,14,11

A= 3,7,9,11,14,12

increment i (now is i=4)

---

find the smallest item by checking all items between i=4 and i=5 (inclusive)
and swap the smallest item with the item at at i=4

A= 3,7,9,11,14,12

A= 3,7,9,11,12,14

increment i (now is i=5)

---

the second last item is now sorted and this means that the last item is also sorted.
So the sort is complete

Bubble Sort Example

A = 12,9,3,7,14,11

i = 0 , 1 , 2 , 3 , 4 , 5

for i= 0 to i=4:
if A[i+1] is less that A[i] swap the two items

i =0
A = 12,9,3,7,14,11

i =1
A = 9,12,3,7,14,11

i =2
A = 9,3,12,7,14,11

i =3
A = 9,3,7,12,14,11

i =4
A = 9,3,7,12,14,11

A = 9,3,7,12,11,14 *end of first pass (the last item is certainly in the right place)*

---

for i= 0 to i=3:
if A[i+1] is less that A[i] swap the two items

i =0
A = 9,3,7,12,11,14

i =1
A = 3,9,7,12,11,14

i =2
A = 3,7,9,12,11,14

i =3
A = 3,7,9,12,11,14 *end of the second pass (the last and second last items are certainly in the right place).*

A = 3,7,9,11,12,14 *We can note that the array is in fact sorted but the algorithm will need another pass to "know" that it has been sorted.*

---

for i= 0 to i=2:
if A[i+1] is less that A[i] swap the two items

i =0
A = 3,7,9,11,12,14

i =1
A = 3,7,9,11,12,14

i =2
A = 3,7,9,11,12,14

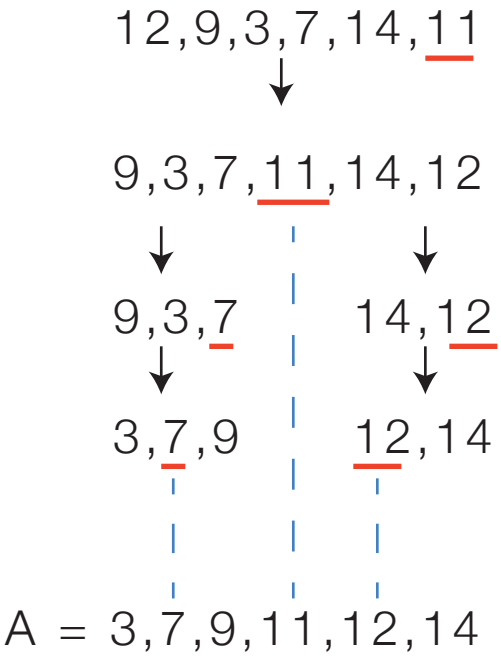
A = 3,7,9,11,12,14 *end of the third pass.*
No swaps have occurred so the array must be sorted

Concept of QuickSort Example

A = 12,9,3,7,14,11

i = 0 , 1 , 2 , 3 , 4 , 5

choose the last item of the array as the "pivot".
arrange the list so that all items to the left of the pivot are less than (or =) the pivot and all items to the right of the pivot are greater than the pivot.



A = 3,7,9,11,12,14

partition the array into sub-divisions above and below the pivot

for each sub-division, choose the last item as the "pivot".
arrange each sub-divided array so that all items to the left of the pivot are less than (or =) the pivot and all items to the right of the pivot are greater than the pivot.

the process is continued unless the subdivision is size = 1