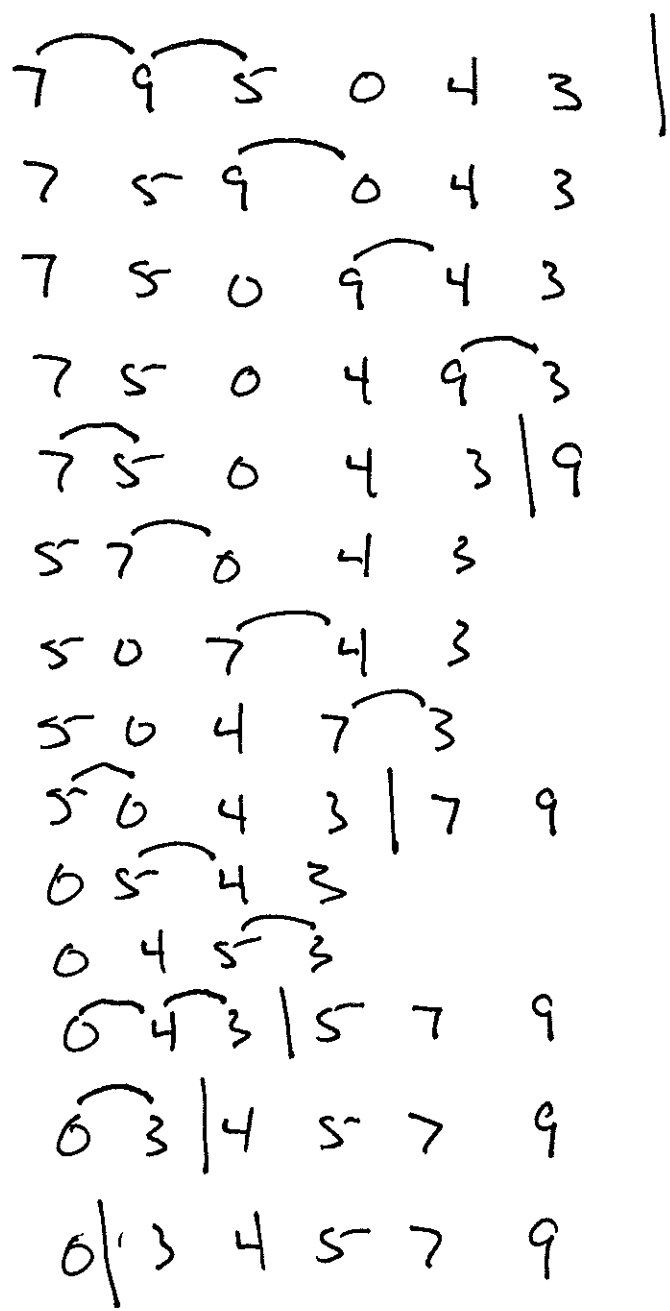


Bubble Sort:

Ex. $n=6$



R

6

5

4

3

2

1

Input: $n \geq 1, a_1, \dots, a_n$

Output: modified list in inc. order

1.) $R \leftarrow n$

2.) while $R \geq 2$

3.) $j \leftarrow 2$

4.) while $j \leq R$

5.) if $a_j < a_{j-1}$

$a_j \leftrightarrow a_{j-1}$ (swap)

7.) $j \leftarrow j + 1$

8.) $R \leftarrow R - 1$

9.) stop

BASIC DA



(6.1)	$temp \leftarrow a_j$
(6.2)	$a_j \leftarrow a_{j-1}$
(6.3)	$a_{j-1} \leftarrow temp$

Refinement of (6)

Run Time

Count # of Comparisons of list elements in best, worst, & avg. cases.

<u>outer</u>	<u>inner</u>	<u># Comp</u>
$R = n$	$2 \leq j \leq n$	$n - 1$
$R = n - 1$	$2 \leq j \leq n - 1$	$n - 2$
$R = n - 2$	$2 \leq j \leq n - 2$	$n - 3$
\vdots		
$R = 3$	$2 \leq j \leq 3$	2
$R = 2$	$2 \leq j \leq 2$	1

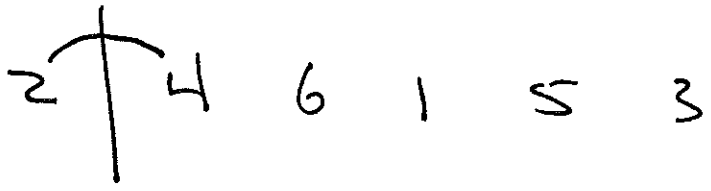
total # of comp = $1 + 2 + \dots + n - 1$
 $= \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$

Insertion Sort

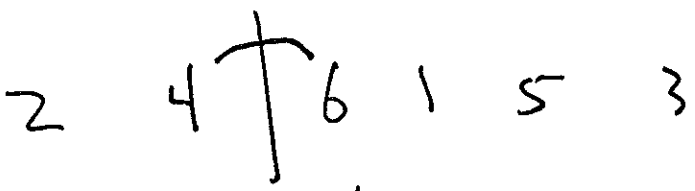
index of leftmost
elt. in unsorted sec.

Ex. $n = 6$

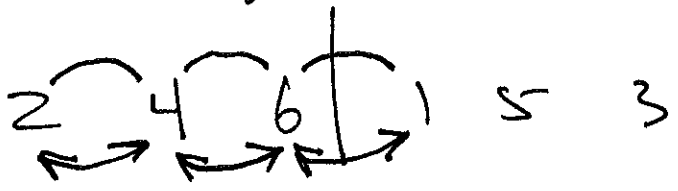
2



2



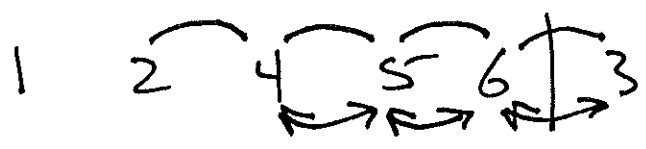
3



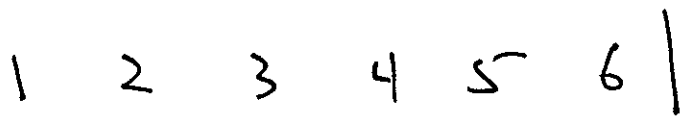
4



5



6



7

Insertion Sort

5

1.) $L \leftarrow 2$

BASIC OP.

2.) while $L \leq n$

3.) $j \leftarrow L$

4.) while $j \geq 2$ and $a_j < a_{j-1}$

5.) $a_j \leftrightarrow a_{j-1}$ (swap)

6.) $j \leftarrow j - 1$

7.) $L \leftarrow L + 1$

8.) stops

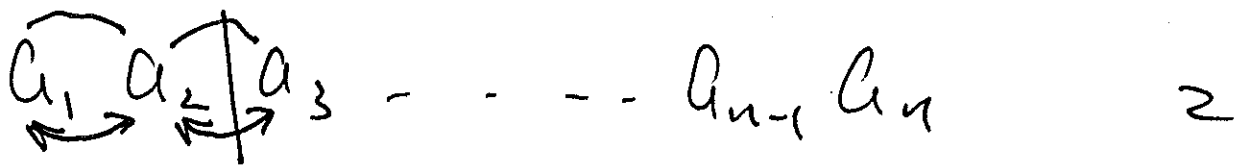
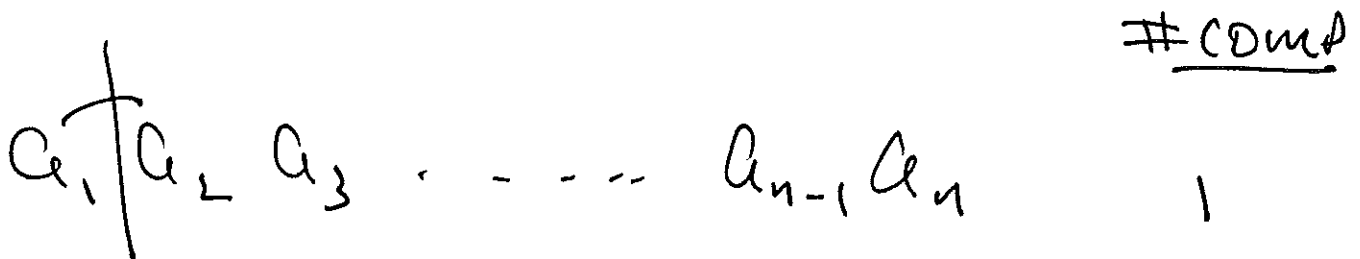
Runtime:

Best case: assume a_1, \dots, a_n is already sorted.

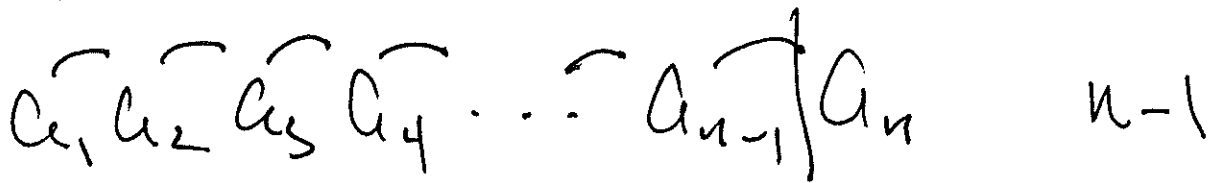
	<u>#comp</u>
$a_1 a_2 \ a_3 \ \dots \ a_{n-1} \ a_n$	1
$a_1 \ a_2 a_3 \ \dots \ a_{n-1} \ a_n$	1
$a_1 \ a_2 \ a_3 a_4 \ \dots \ a_{n-1} \ a_n$	1
⋮	
$a_1 \ a_2 \ a_3 \ a_4 \ \dots \ a_{n-1} a_n$	<u>1</u>

Best case #comp = $n-1$

Worst Case: Assume a_1, \dots, a_n is sorted in descending (rather than asc.) order.



⋮



Worst case # comps =

$$= 1 + 2 + 3 + \dots + (n-1) = \frac{n(n-1)}{2}$$

$$= \frac{1}{2} n^2 - \frac{1}{2} n$$

Average Case! ? ?

will Determine this
experimentally in future
lab assignment