

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← i
  j ← i + 1
  while j < len(A)
    if A[j] < A[loc]
      loc ← j
    j ← j + 1
  temp ← A[loc]
  A[loc] ← A[i]
  A[i] ← temp
  i ← i + 1
```

### INSERTION SORT (A)

```
i ← 1
while i < len(A)
  j ← i
  while j > 0 and A[j] < A[j - 1]
    temp ← A[j - 1]
    A[j - 1] ← A[j]
    A[j] ← temp
  j ← j - 1
  i ← i + 1
```

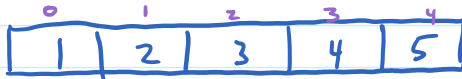
PRECONDITIONS:  $A$  is an array of mutually comparable items

POSTCONDITIONS:  $A$  is a permutation of  $A_{in}$  with  
 $A[0] \leq A[1] \leq \dots \leq A[\text{len}(A) - 1]$

## Selection Sort and Insertion Sort

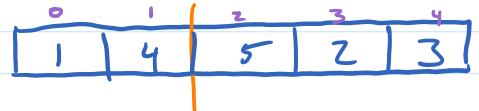
### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← i
  j ← i + 1
  while j < len(A)
    if A[j] < A[loc]
      loc ← j
    j ← j + 1
  temp ← A[loc]
  A[loc] ← A[i]
  A[i] ← temp
  i ← i + 1
```



### INSERTION SORT (A)

```
i ← 1
while i < len(A)
  j ← i
  while j > 0 and A[j] < A[j - 1]
    temp ← A[j - 1]
    A[j - 1] ← A[j]
    A[j] ← temp
  j ← j - 1
  i ← i + 1
```



## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← i
  j ← i + 1
  while j < len(A)
    if A[j] < A[loc]
      loc ← j
    j ← j + 1
  temp ← A[loc]
  A[loc] ← A[i]
  A[i] ← temp
  i ← i + 1
```

0	1	2	3	4
1	2	3	4	5

### INSERTION SORT (A)

```
i ← 1
while i < len(A)
  j ← i
  while j > 0 and A[j] < A[j - 1]
    temp ← A[j - 1]
    A[j - 1] ← A[j]
    A[j] ← temp
  j ← j - 1
  i ← i + 1
```

0	1	2	3	4
1	2	3	4	5

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

$i \leftarrow 0$

while  $i < \text{len}(A) - 1$

$\text{loc} \leftarrow \text{findMin}(A, i)$

$\text{swap}(A, \text{loc}, i)$

$i \leftarrow i + 1$

PRE:  $0 \leq \text{loc}, i < \text{len}(A)$   
POST:  $A[k] = A_{\text{in}}[k]$  for  $0 \leq k < \text{len}(A)$ ,  
 $k \neq \text{loc}$  and  $k \neq i$

$A[\text{loc}] = A_{\text{in}}[i]$

$A[i] = A_{\text{in}}[\text{loc}]$

### INSERTION SORT (A)

$i \leftarrow 1$

while  $i < \text{len}(A)$

$j \leftarrow i$

  while  $j > 0$  and  $A[j] < A[j-1]$

$\text{temp} \leftarrow A[j-1]$

$A[j-1] \leftarrow A[j]$

$A[j] \leftarrow \text{temp}$

$j \leftarrow j - 1$

$i \leftarrow i + 1$

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← findMin(A, i)
  swap(A, loc, i)
  i ← i + 1
```

### INSERTION SORT (A)

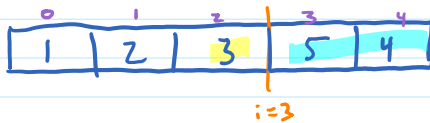
```
i ← 1
while i < len(A)
  insert(A, i)
  i ← i + 1
```

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← findMin(A, i)
  swap(A, loc, i)
  i ← i + 1
```

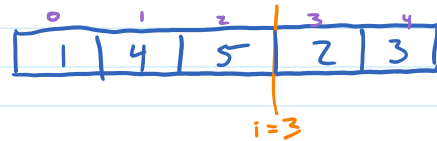
INVARIANT: a)  $i = n$   
b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$   
c)  $i = 0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$



### INSERTION SORT (A)

```
i ← 1
while i < len(A)
  insert(A, i)
  i ← i + 1
```

INVARIANT: a)  $i = n + 1$   
b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$



## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0  
→ while i < len(A) - 1  
    loc ← findMin(A, i)  
    swap(A, loc, i)  
    i ← i + 1
```

INVARIANT: a)  $i = n$

b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$

c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d) A is a permutation of  $A_{in}$

Basis ( $n=0$ ): a) initialization makes  $i=0=n$  so  $i=n$

b) vacuously true for all  $k$  such that  $1 \leq k \leq i-1$ ,  $A[k-1] \leq A[k]$

c)  $i=0$ , so true

d)  $A = A_{in}$

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

$i \leftarrow 0$   
→ while  $i < \text{len}(A) - 1$   
     $\text{loc} \leftarrow \text{findMin}(A, i)$   
     $\text{swap}(A, \text{loc}, i)$   
     $i \leftarrow i + 1$

INVARIANT: a)  $i \leq n$

b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$

c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d) A is a permutation of  $A_{in}$

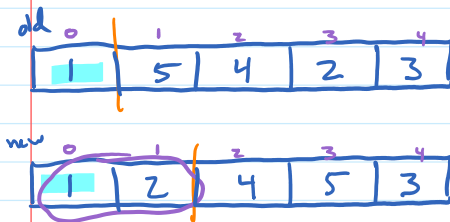
Induction: Suppose INV is true after  $n$  iterations and  $i < \text{len}(A) - 1$  [want INV true after  $n+1$  iterations]

b) Want:  $A_{new}[0] \leq \dots \leq A_{new}[i_{new}-1]$

If  $i_{new} = 1$  then vacuously true

Suppose  $i_{new} > 1$  (and so  $i_{old} = i_{new} - 1 > 0$ )

$A_{new}[0] \leq \dots \leq A_{new}[i_{old}-1]$





## Selection Sort and Insertion Sort

### SELECTION SORT (A)

$i \leftarrow 0$   
→ while  $i < \text{len}(A) - 1$   
   $\text{loc} \leftarrow \text{findMin}(A, i)$   
   $\text{swap}(A, \text{loc}, i)$   
   $i \leftarrow i + 1$

INVARIANT: a)  $i \leq n$   
b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$   
c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d) A is a permutation of  $A_{in}$

Induction: Suppose INV is true after  $n$  iterations and  $i < \text{len}(A) - 1$  [want INV true after  $n+1$  iterations]

b) Want:  $A_{new}[0] \leq \dots \leq A_{new}[i_{new}-1]$

If  $i_{new} = 1$  then vacuously true

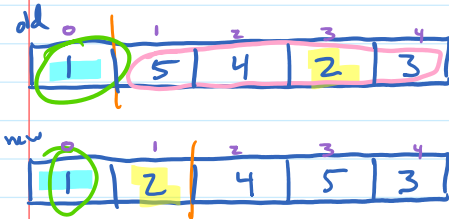
Suppose  $i_{new} > 1$  (and so  $i_{old} = i_{new} - 1 > 0$ )

$A_{new}[0] \leq \dots \leq A_{new}[i_{old}-1]$

$A_{new}[i_{old}] = A_{old}[\text{loc}] = \min(A_{old}[i_{old}], \dots, A_{old}[\text{len}(A)-1])$

$A_{new}[i_{old}-1] \leq A_{old}[\text{loc}]$

$A_{new}[0] \leq \dots \leq A_{new}[i_{new}-2] \leq A_{old}[\text{loc}] = A_{new}[i_{new}-1]$



## Selection Sort and Insertion Sort

### SELECTION SORT (A)

$i \leftarrow 0$   
→ while  $i < \text{len}(A) - 1$   
     $\text{loc} \leftarrow \text{findMin}(A, i)$   
     $\text{swap}(A, \text{loc}, i)$   
     $i \leftarrow i + 1$

INVARIANT: a)  $i \leq n$

b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$

c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d)  $A$  is a permutation of  $A_{in}$

Induction: Suppose INV is true after  $n$  iterations and  $i < \text{len}(A) - 1$  [want INV true after  $n+1$  iterations]

c) want:  $A_{new}[\text{loc}] \leq A_{new}[\text{loc}+1], \dots, A_{new}[\text{len}(A)-1]$

$A_{new}[\text{loc}] = A_{old}[\text{loc}] = \min(A_{old}[\text{loc}], \dots, A_{old}[\text{len}(A)-1])$

$A_{new}[\text{loc}] \leq A_{old}[\text{loc}], \dots, A_{old}[\text{len}(A)-1]$

$A_{new}[\text{loc}-1] \leq A_{new}[k]$  for all  $k$ ,  $\text{loc} \leq k < \text{len}(A)$  and  $k \neq \text{loc}$ .  
 $A_{new}[\text{loc}] \leq A_{old}[\text{loc}] = A_{new}[\text{loc}]$

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0  
→ while i < len(A) - 1  
    loc ← findMin(A, i)  
    swap(A, loc, i)  
    i ← i + 1
```

INVARIANT: a)  $i \leq n$

b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$

c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d) A is a permutation of  $A_{in}$

Induction: Suppose INV is true after  $n$  iterations and  $i < \text{len}(A) - 1$  [want INV true after  $n+1$  iterations]

d)  $A_{old}$  is a permutation of  $A_{in}$   
 $A_{new}$  is a permutation of  $A_{old}$

so  $A_{new}$  is a permutation of  $A_{in}$

## Selection Sort and Insertion Sort

### SELECTION SORT (A)

```
i ← 0
while i < len(A) - 1
  loc ← findMin(A, i)
  swap(A, loc, i)
  i ← i + 1
```

INVARIANT: a)  $i = n$

b)  $A[0] \leq A[1] \leq \dots \leq A[i-1]$

c)  $i=0$  or  $A[i-1] \leq A[i], \dots, A[\text{len}(A)-1]$

d) A is a permutation of  $A_{in}$

Termination: when  $n < \text{len}(A) - 1$ ,  $i < \text{len}(A) - 1$

when  $n = \text{len}(A) - 1$ ,  $i = \text{len}(A) - 1$

Postcondition: when  $n = \text{len}(A) - 1$ ,  $i = n = \text{len}(A) - 1$

$A[0] \leq \dots \leq A[\text{len}(A)-2]$

$A[\text{len}(A)-2] \leq A[\text{len}(A)-1]$

A is a permutation of  $A_{in}$

$A[0] \leq \dots \leq A[\text{len}(A)-2] \leq A[\text{len}(A)-1]$