

# Selection vs. Insertion Sort

Slides by **Sean Szumlanski**  
for **CS106B**, Programming Abstractions

*Winter 2024*

# Selection Sort

(select the smallest element)

# Insertion Sort

(insert into sorted partition)

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

# Selection Sort

(select the smallest element)

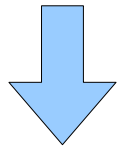
- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

MIN: 10



0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

# Selection Sort

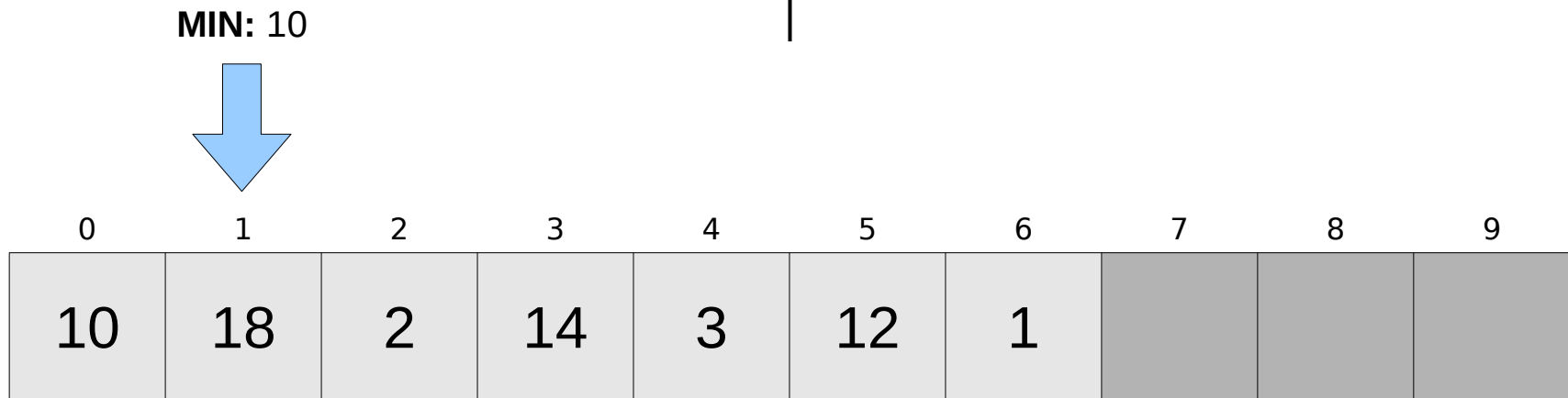
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

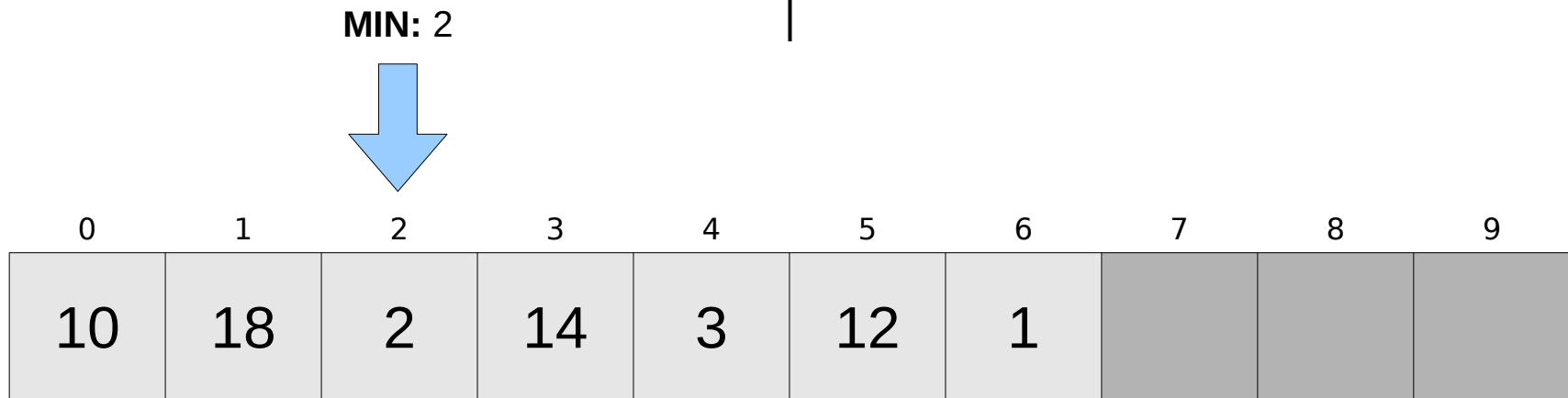
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy





# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

(select the smallest element)

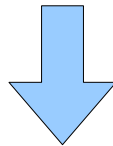
- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

MIN: 2



0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

# Selection Sort

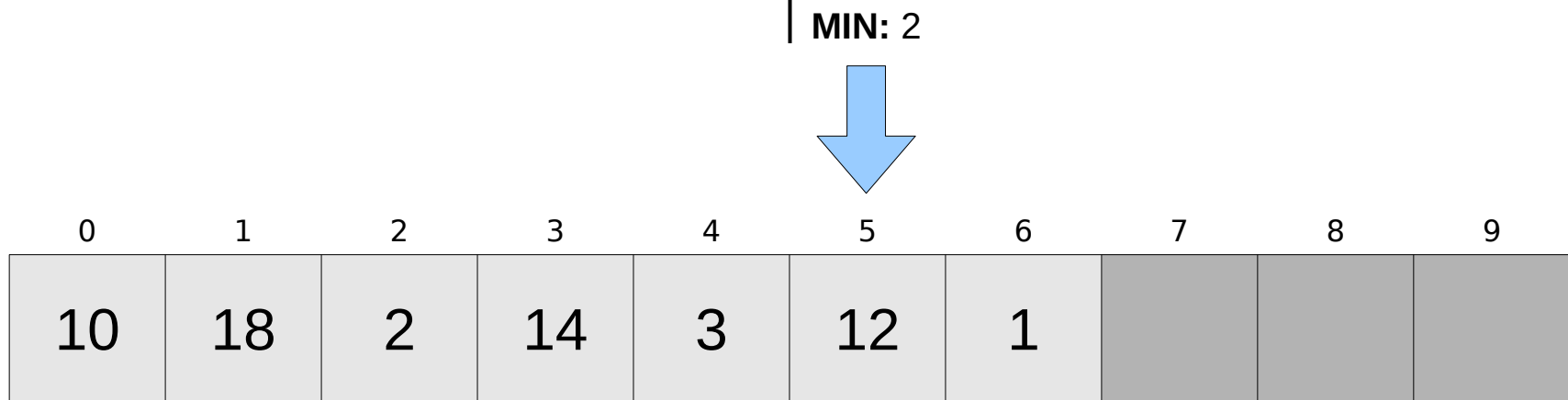
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

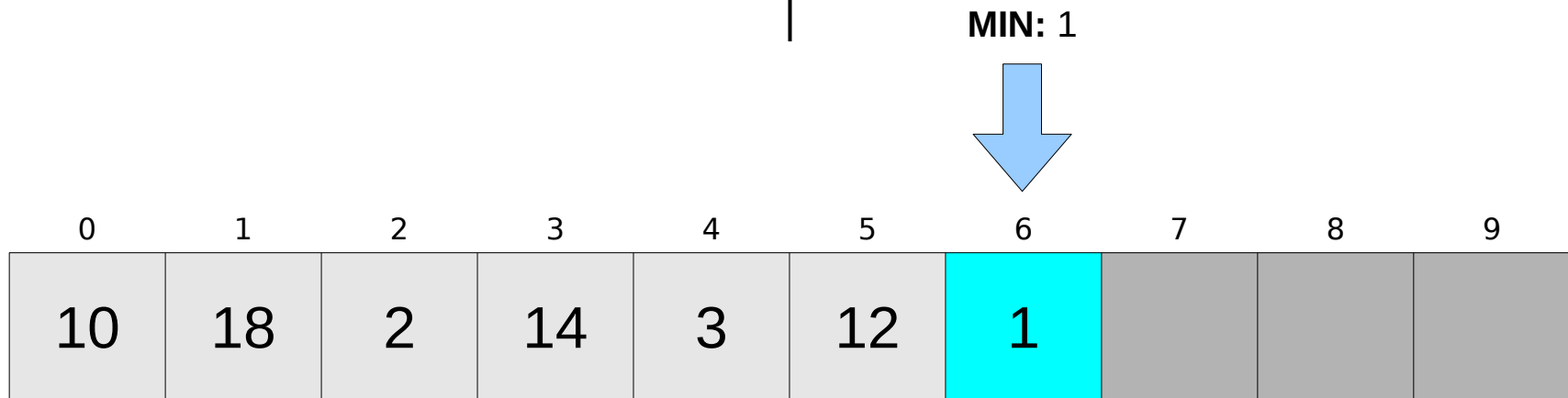
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

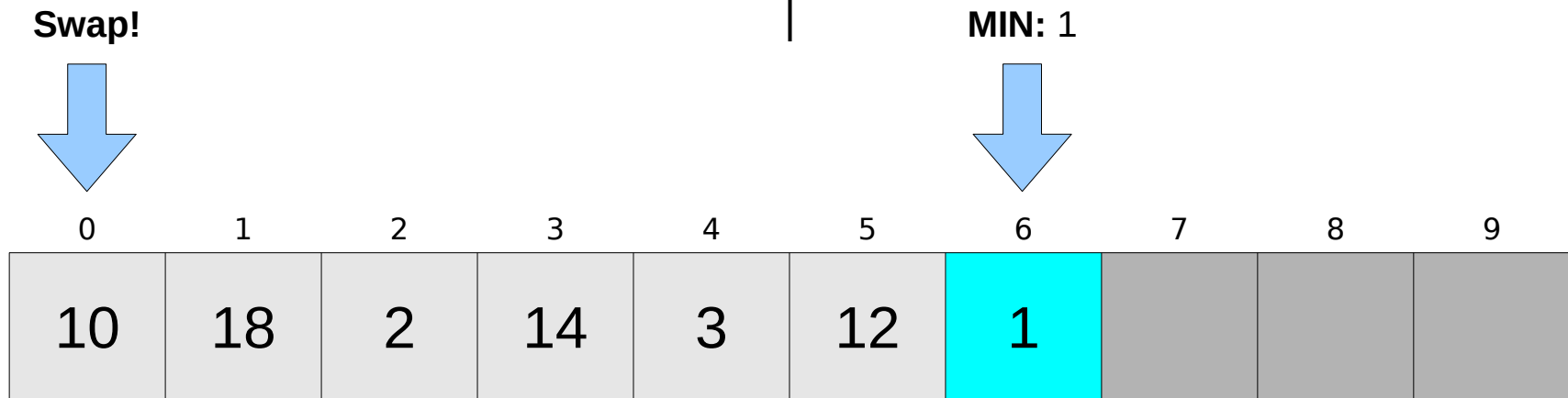
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

(select the smallest element)

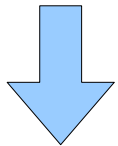
- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

MIN: 1



0	1	2	3	4	5	6	7	8	9
1	18	2	14	3	12	10			

# Selection Sort

(select the smallest element)

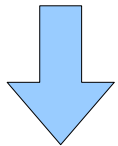
- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

MIN: 1



0	1	2	3	4	5	6	7	8	9
1	18	2	14	3	12	10			



# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

0	1	2	3	4	5	6	7	8	9
1	18	2	14	3	12	10			

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

0	1	2	3	4	5	6	7	8	9
1	12	14	18	21	3	10			

# Selection Sort

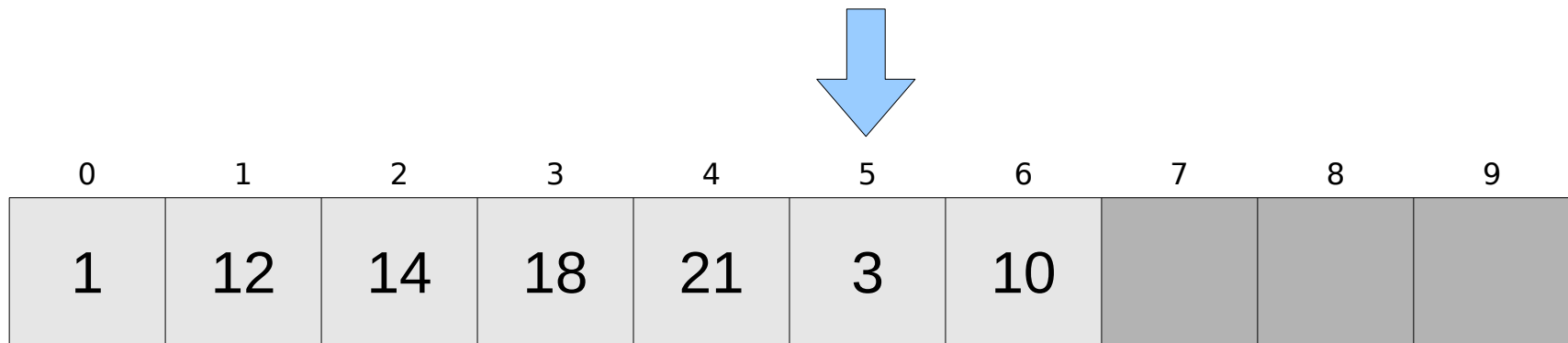
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

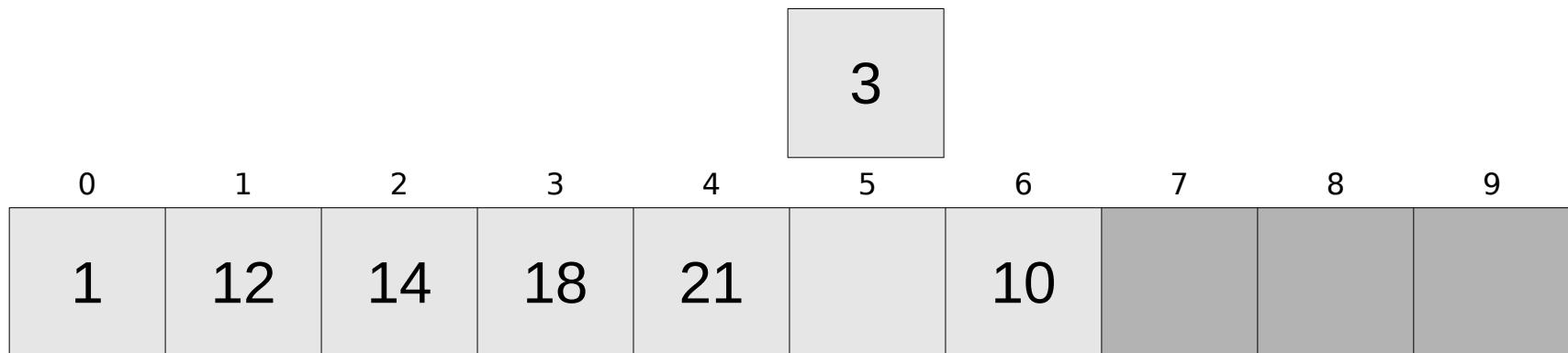
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

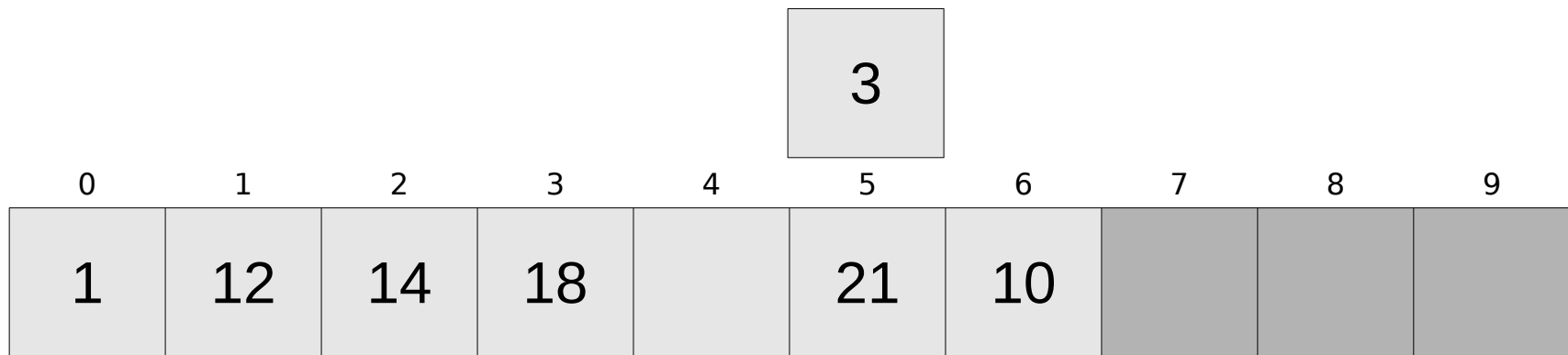
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



Swap!

# Selection Sort

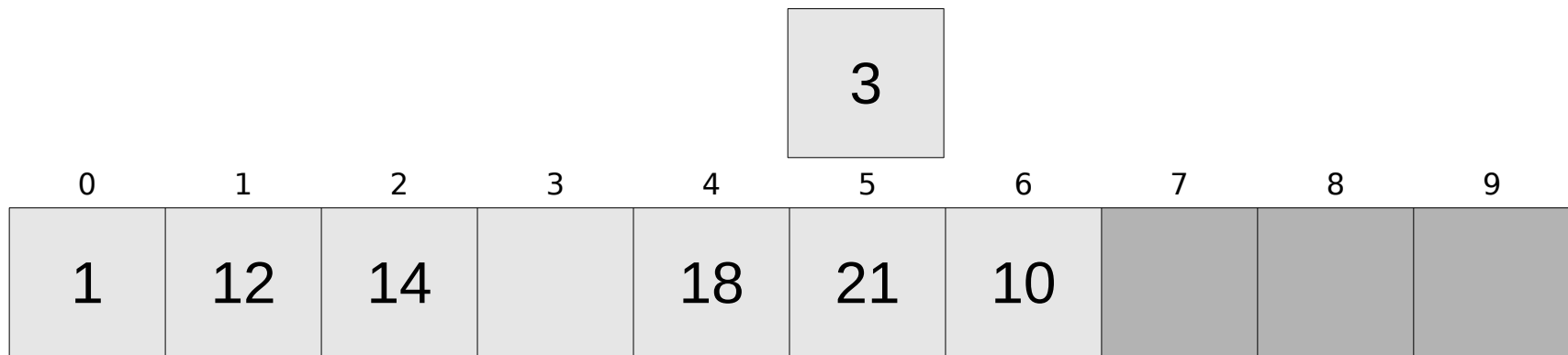
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



Swap!

# Selection Sort

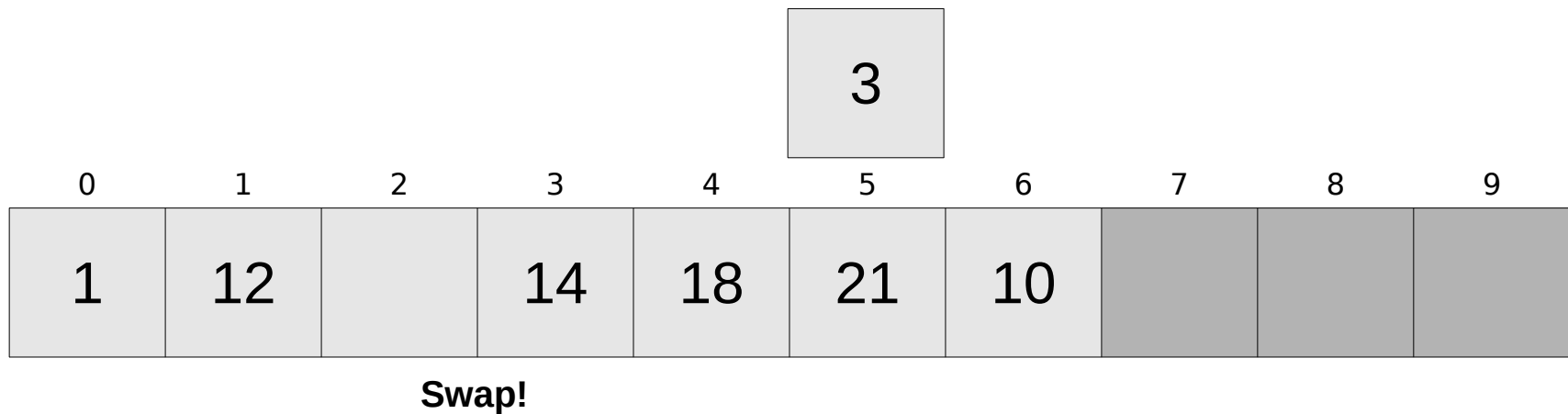
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy





# Selection Sort

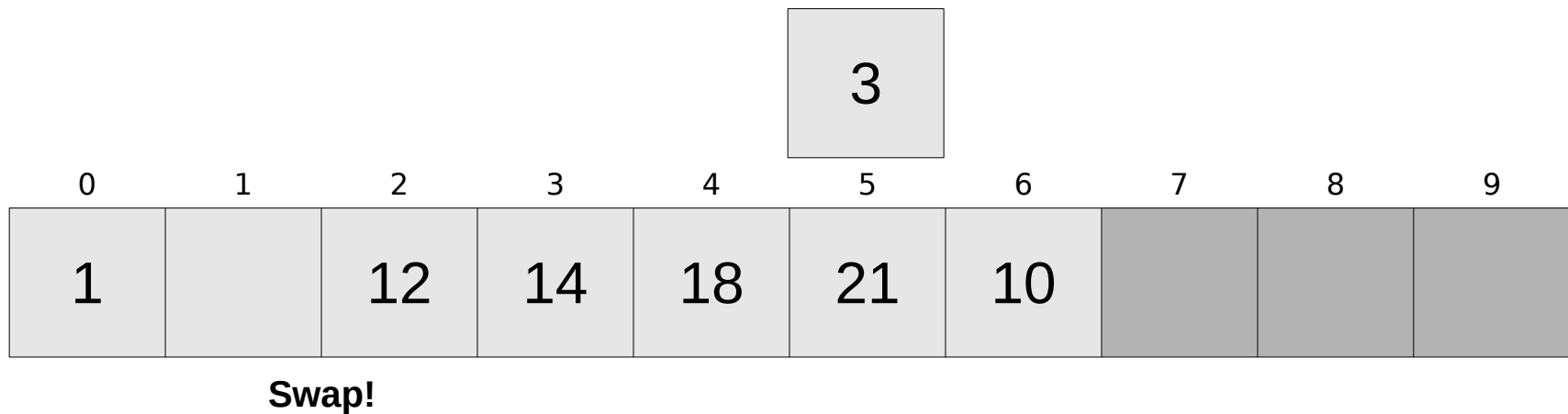
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



# Selection Sort

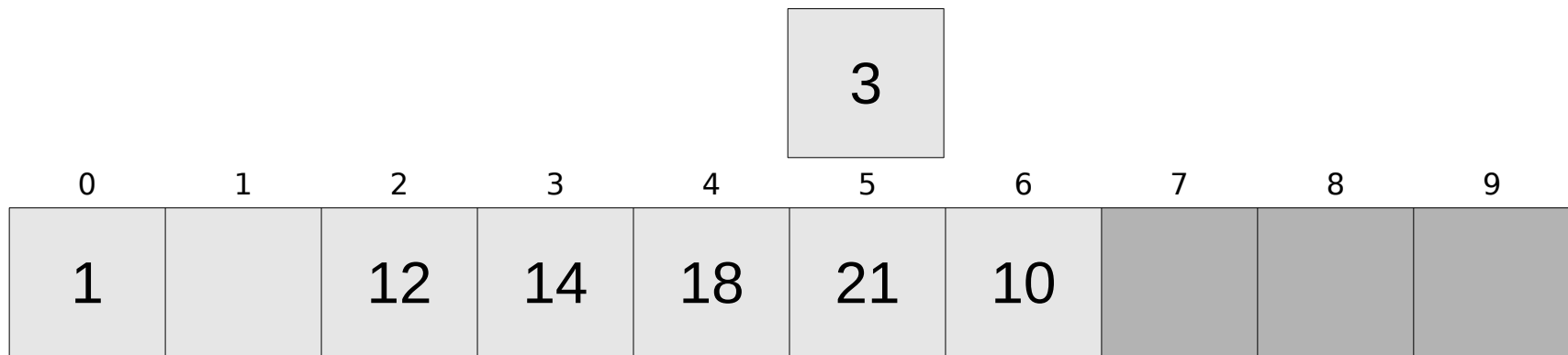
(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy



**Stop!**

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

0	1	2	3	4	5	6	7	8	9
1	3	12	14	18	21	10			

**Stop!**

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

0	1	2	3	4	5	6	7	8	9
1	3	12	14	18	21	10			

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting fridges in a warehouse by price.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting fridges in a warehouse by price.

- Comparisons are easy.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting fridges in a warehouse by price.

- Comparisons are easy.
- Swapping is obnoxious.



# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting fridges in a warehouse by price.

**Selection  
Sort!**

- Comparisons are easy.
- Swapping is obnoxious.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Athletic competition with 1-on-1 comparison.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Athletic competition with 1-on-1 comparison.

- Comparisons are taxing.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Athletic competition with 1-on-1 comparison.

- Comparisons are taxing.
- Swapping is easy.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Athletic competition with 1-on-1 comparison.

- Comparisons are taxing.
- Swapping is easy.

***Insertion  
Sort!***

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Computational biology with simulations.



# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Computational biology with simulations.

- Comparisons are expensive.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Computational biology with simulations.

- Comparisons are expensive.
- Swapping is easy (unique IDs?).

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Computational biology with simulations.

- Comparisons are expensive.
- Swapping is easy (unique IDs?).

***Insertion  
Sort!***

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting data-heavy student records.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting data-heavy student records.

- Comparisons are easy.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting data-heavy student records.

- Comparisons are easy.
- Swapping is expensive.

# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

**Example:** Sorting data-heavy student records.

**Selection  
Sort!**

- Comparisons are easy.
- Swapping is expensive.



# Selection Sort

(select the smallest element)

- Comparison Heavy

# Insertion Sort

(insert into sorted partition)

- Swap heavy

# Selection Sort

(select the smallest element)

- Comparison Heavy
  - Sorting refrigerators
  - Sorting data-heavy student records

# Insertion Sort

(insert into sorted partition)

- Swap heavy
  - Athletic comparisons
  - Expensive simulations

# Selection Sort

(select the smallest element)

- Comparison Heavy
  - Sorting refrigerators
  - Sorting data-heavy student records
- Runtime consistency

# Insertion Sort

(insert into sorted partition)

- Swap heavy
  - Athletic comparisons
  - Expensive simulations

# Selection Sort

(select the smallest element)

- Comparison Heavy
  - Sorting refrigerators
  - Sorting data-heavy student records
- Runtime consistency

# Insertion Sort

(insert into sorted partition)

- Swap heavy
  - Athletic comparisons
  - Expensive simulations
- Faster in special cases

# Selection Sort

(select the smallest element)

- Comparison Heavy
  - Sorting refrigerators
  - Sorting data-heavy student records
- Runtime consistency
- Ease of coding and debugging

# Insertion Sort

(insert into sorted partition)

- Swap heavy
  - Athletic comparisons
  - Expensive simulations
- Faster in special cases

# Selection Sort

(select the smallest element)

- Comparison Heavy
  - Sorting refrigerators
  - Sorting data-heavy student records
- Runtime consistency
- Ease of coding and debugging

# Insertion Sort

(insert into sorted partition)

- Swap heavy
  - Athletic comparisons
  - Expensive simulations
- Faster in special cases
- Potential performance gains, but potentially harder to code