CS 2050
Computer Science II

Thyago Mota

METROPOLITAN
STATE UNIVERSITY℠
OF DENVER
LIVES TRANSFORMED

Agenda

- Sorting Algorithms:
    - Introduction
    - Selection Sort

Sorting

- In computer science, sorting is the process of arranging elements in a collection in some pre-defined order (e.g., alphabetical or numerical order)

Sorting

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

Sorting

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

↓

```
[1, 3, 10, 12, 13, 27, 34, 56, 77, 79, 84]
```

Selection Sort

- It divides the collection into two parts:
    - one built from left to right with the elements sorted, and
    - the remaining (to be sorted) elements

Selection Sort

- The algorithm works by repeatedly moving the smaller (or larger, depending on the sorting criterion) element from the right part to the left part, until all elements are evaluated

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

↓

```
[][13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

↓

```
[][13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

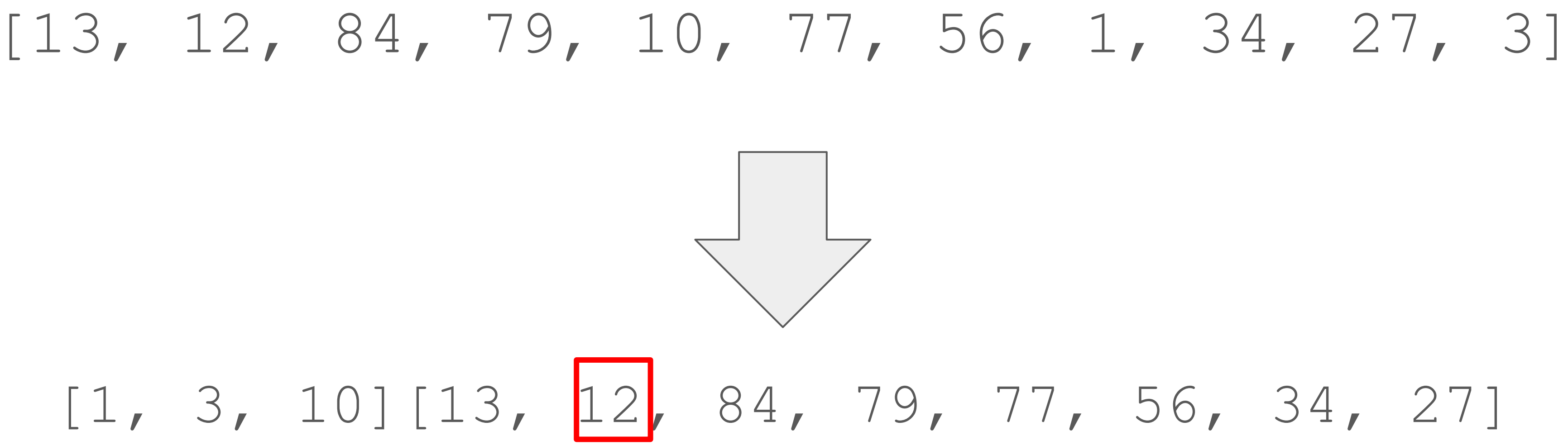
```
[1][13, 12, 84, 79, 10, 77, 56, 34, 27, 3]
```

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

```
[1, 3][13, 12, 84, 79, 10, 77, 56, 34, 27]
```

Selection Sort



```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

```
[1, 3, 10][13, 12, 84, 79, 77, 56, 34, 27]
```

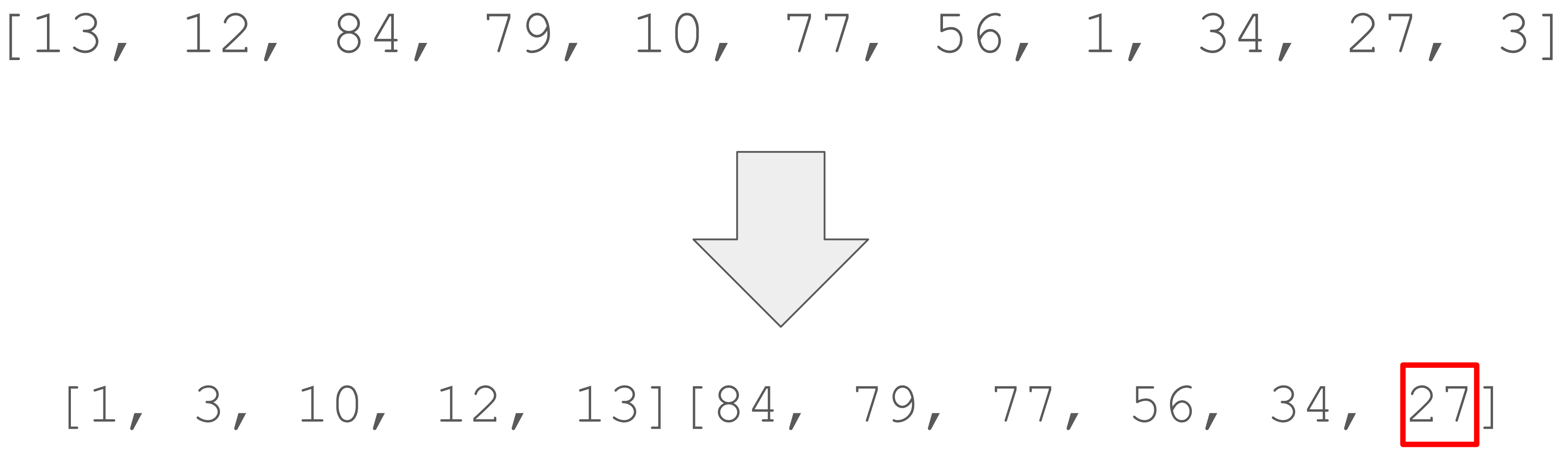
Selection Sort

[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]

↓

[1, 3, 10, 12][$\boxed{13}$, 84, 79, 77, 56, 34, 27]

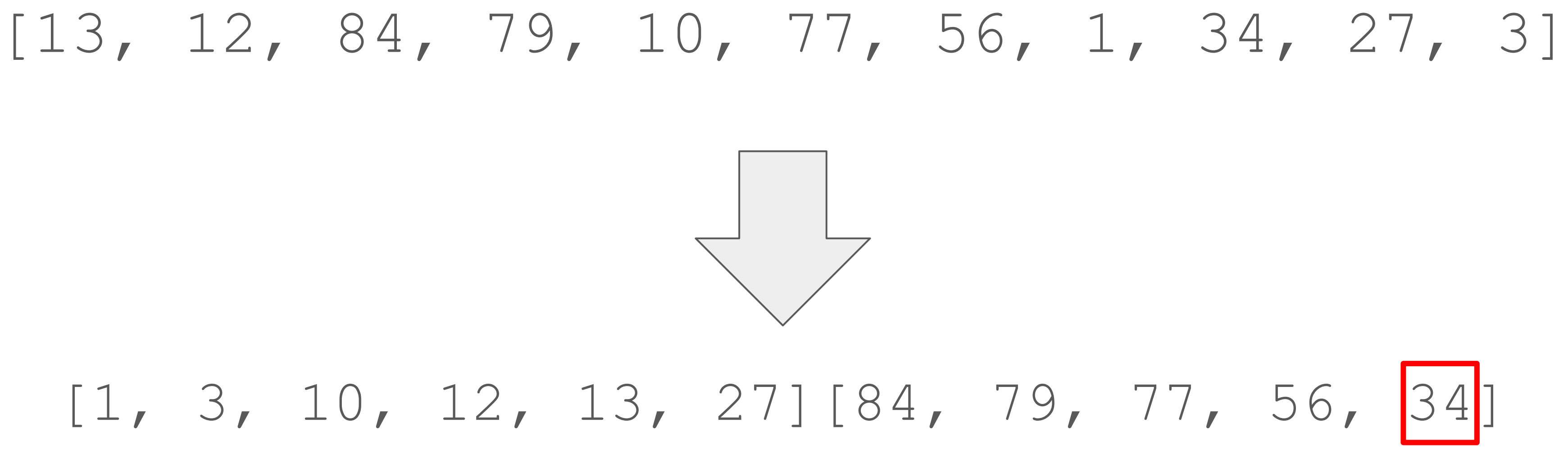
Selection Sort



```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

```
[1, 3, 10, 12, 13][84, 79, 77, 56, 34, 27]
```

Selection Sort



[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]

[1, 3, 10, 12, 13, 27][84, 79, 77, 56, 34]

Selection Sort

[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]

⇩

[1, 3, 10, 12, 13, 27, 34][84, 79, 77, $\boxed{56}$]

Selection Sort

[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]

↓

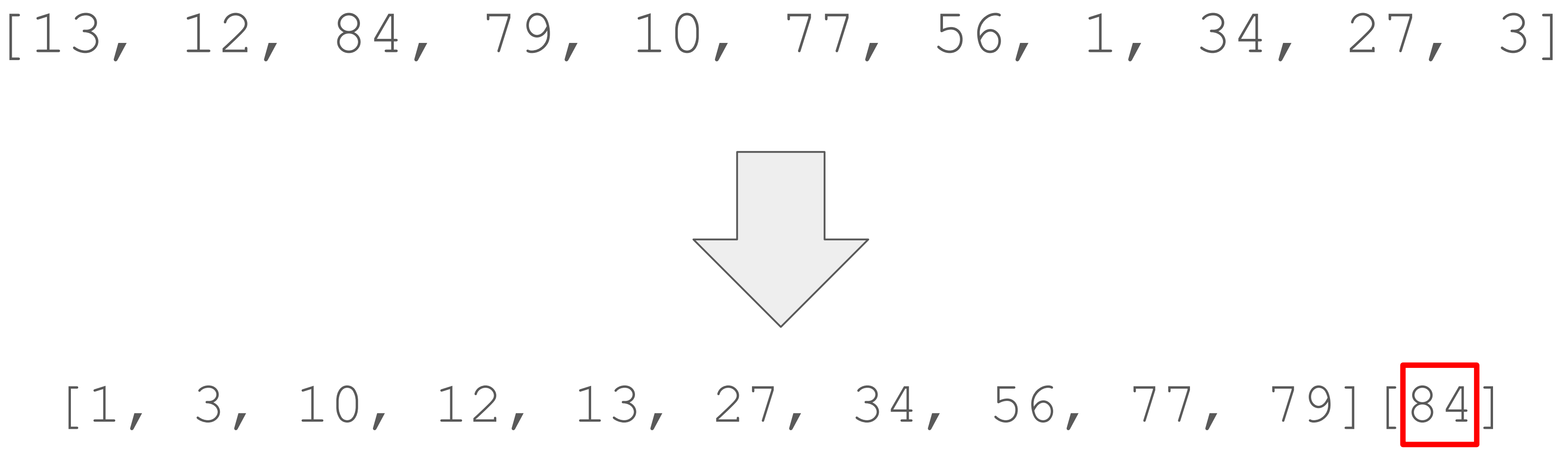
[1, 3, 10, 12, 13, 27, 34, 56][84, 79, $\boxed{77}$]

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

```
[1, 3, 10, 12, 13, 27, 34, 56, 77][84, 79]
```

Selection Sort



```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

```
[1, 3, 10, 12, 13, 27, 34, 56, 77, 79][84]
```

Selection Sort

```
[13, 12, 84, 79, 10, 77, 56, 1, 34, 27, 3]
```

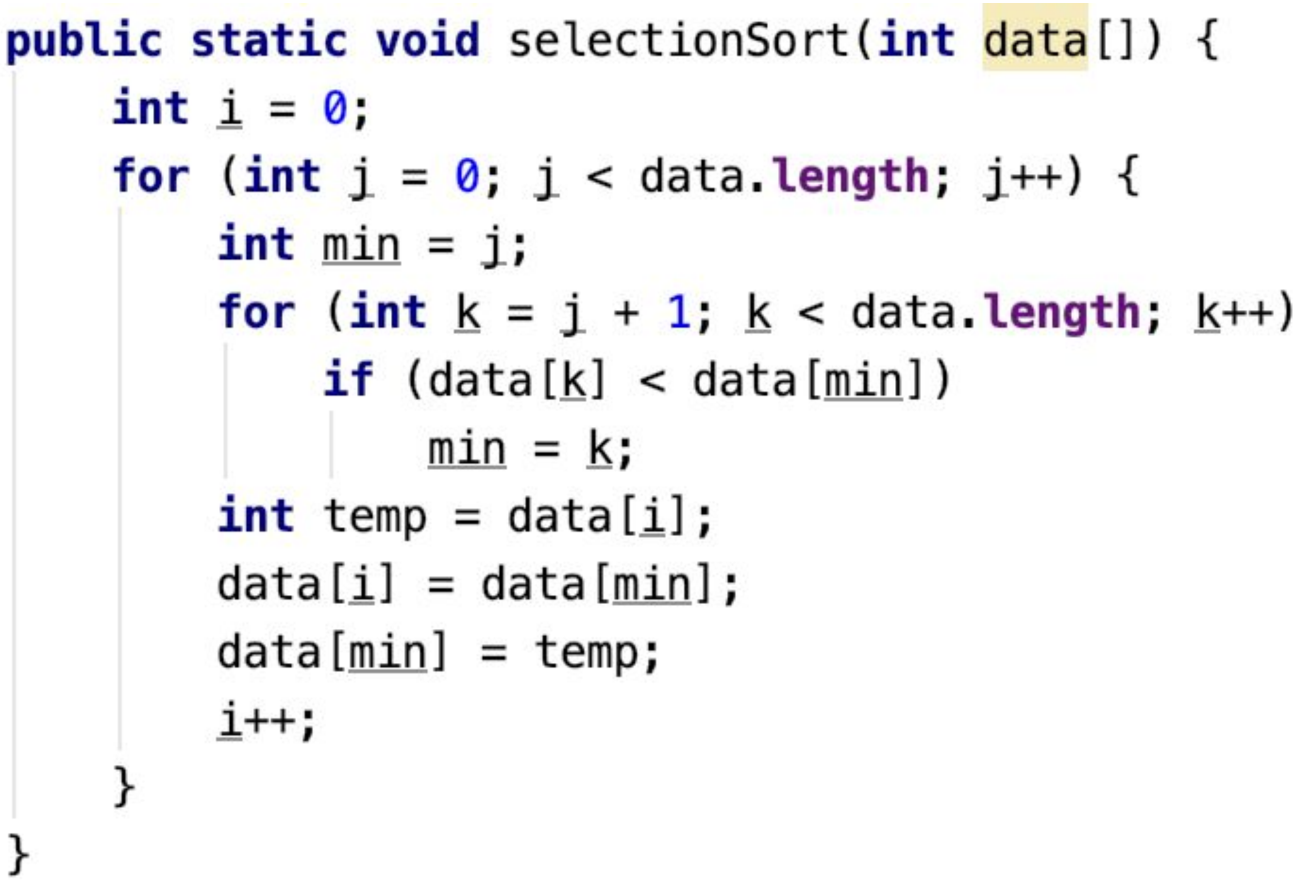
```
[1, 3, 10, 12, 13, 27, 34, 56, 77, 79, 84][]
```

Selection Sort

Pause the video now and try to implement the selection sort algorithm!

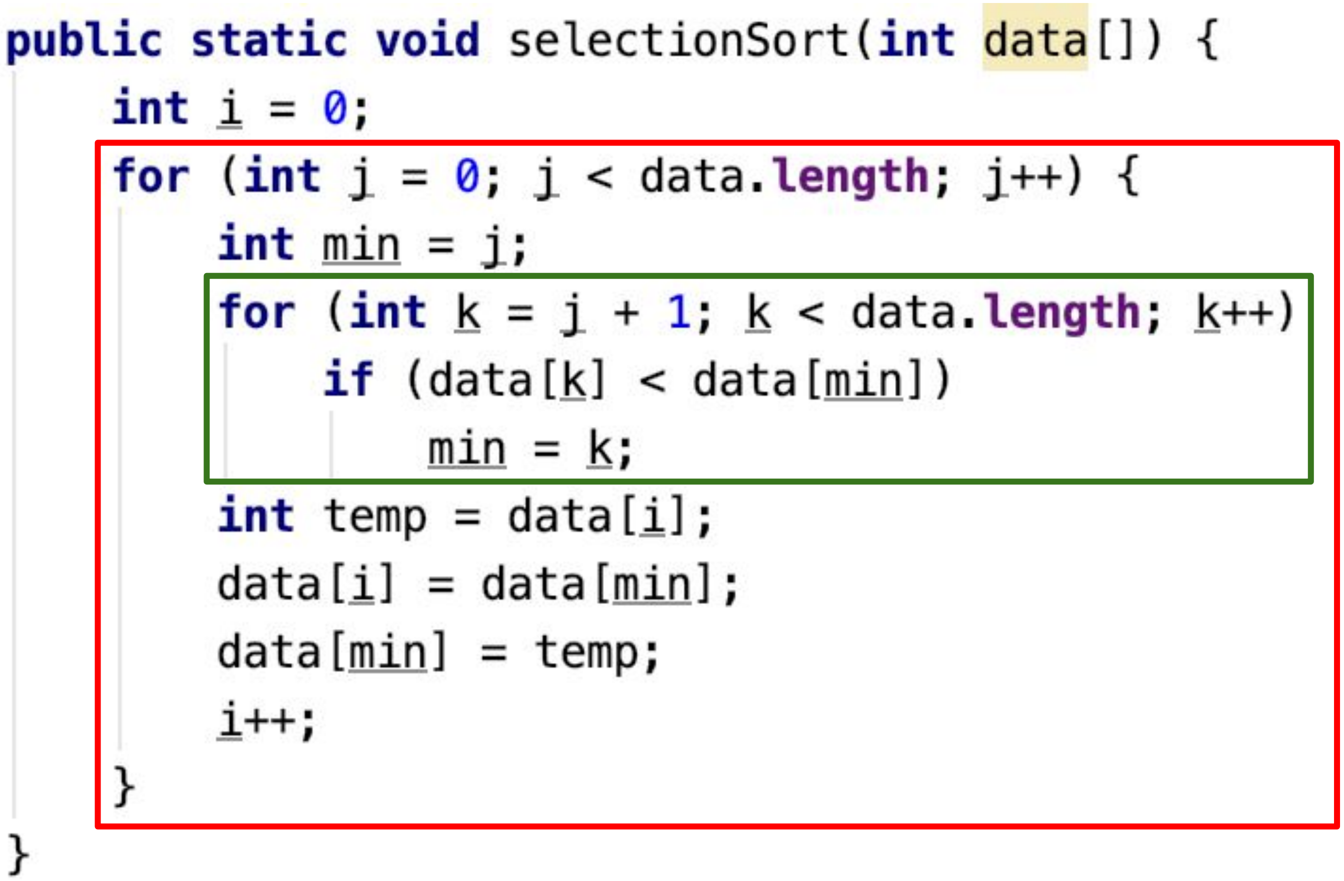
GitHub

Selection Sort



```java
public static void selectionSort(int data[]) {
    int i = 0;
    for (int j = 0; j < data.length; j++) {
        int min = j;
        for (int k = j + 1; k < data.length; k++)
            if (data[k] < data[min])
                min = k;
        int temp = data[i];
        data[i] = data[min];
        data[min] = temp;
        i++;
    }
}
```

Selection Sort



```java
public static void selectionSort(int data[]) {
    int i = 0;
    for (int j = 0; j < data.length; j++) {
        int min = j;
        for (int k = j + 1; k < data.length; k++)
            if (data[k] < data[min])
                min = k;
        int temp = data[i];
        data[i] = data[min];
        data[min] = temp;
        i++;
    }
}
```

$O(n^2)$