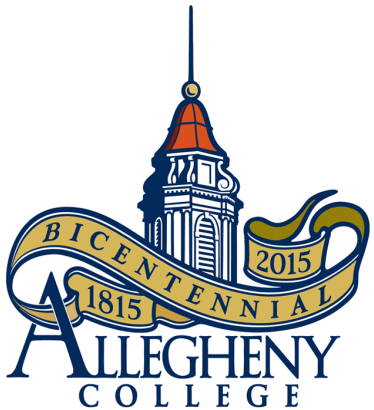
# *CS202 - Algorithm Analysis*

# Selection Sort

Aravind Mohan

Allegheny College

March 24, 2021



# Discussion Based On ...

**Sedgewick 1.3, Queues, 2.3 Selection Sort**

# Selection Sort Algorithm

**Strategy:**

- Place all items in Bag 1.
- Remove the first item from Bag 1 and place it in Bag 2.
- Find the minimum item $(X)$ from Bag 1 and switch it with the most recently placed item $(Y)$ in Bag 2, **if** $X < Y$.
- Repeat steps 2 and 3 until Bag 1 is empty and Bag 2 is full.
- **Note:** Both bags are in one array itself, since this is in-place sorting. The concept of Bag is only logical and not physical.

# Selection Sort Algorithm

**Algorithm -** Selection($A$)

**Input:** an n-element un-sorted array A of integer values.

**Output:** an n-element sorted array A of integer values.

```
1: for i = 0 to n − 1 do
2:     min ← i
3:     for j = i + 1 to n do
4:         if A[j] < A[min] then
5:             min ← j
6:         end if
7:     end for
8:     if min <> i then
9:         swap A[i] & A[min]
10:    end if
11: end for
```

# Selection Sort Example

**Input:** [5,4,3,2,1])

```
i=0       j=[1,2,3,4]      min=4   array=[1, 4, 3, 2, 5]
i=1       j=[2,3,4]        min=3   array=[1, 2, 3, 4, 5]
i=2       j=[3,4]          min=2   array=[1, 2, 3, 4, 5]
i=3       j=[4]    min=3   array=[1, 2, 3, 4, 5]
```

# Selection Sort Example

**Input:** [1,2,3,4,5])

```
i=0     j=[1,2,3,4]     min=0   array=[1, 2, 3, 4, 5]
i=1     j=[2,3,4]       min=1   array=[1, 2, 3, 4, 5]
i=2     j=[3,4]         min=2   array=[1, 2, 3, 4, 5]
i=3     j=[4]   min=3   array=[1, 2, 3, 4, 5]
```

# Selection Sort Example

**Input:** [5,6,9,0,8,2,7,1,3])

```
i=0     j=[1,2,3,4,5,6,7,8]     min=3   array=[0, 6, 9, 5, 8, 2, 7, 1, 3]
i=1     j=[2,3,4,5,6,7,8]       min=7   array=[0, 1, 9, 5, 8, 2, 7, 6, 3]
i=2     j=[3,4,5,6,7,8]         min=5   array=[0, 1, 2, 5, 8, 9, 7, 6, 3]
i=3     j=[4,5,6,7,8]   min=8   array=[0, 1, 2, 3, 8, 9, 7, 6, 5]
i=4     j=[5,6,7,8]     min=8   array=[0, 1, 2, 3, 5, 9, 7, 6, 8]
i=5     j=[6,7,8]       min=7   array=[0, 1, 2, 3, 5, 6, 7, 9, 8]
i=6     j=[7,8]         min=6   array=[0, 1, 2, 3, 5, 6, 7, 9, 8]
i=7     j=[8]   min=8   array=[0, 1, 2, 3, 5, 6, 7, 8, 9]
```

# Worst Case Analysis

- $L_2$, $L_8$ to $L_{10}$ executes one time each for every iteration.

$$1 + 1 + 1 + 1 + \cdots + n - 1 = \textbf{n-1}$$

- $L_4$ to $L_6$ executes in the sequence shown below for all iterations combined.

$$(n-1) + (n-2) + (n-3) + (n-4) + \cdots$$
$$+(n-(n-3)) + (n-(n-2)) + (n-(n-1))$$

$$= (n-1) + (n-2) + (n-3) + (n-4) + \cdots + 3 + 2 + 1$$
$$= \frac{n \times (n-1)}{2} = \frac{n^2}{2} = \textbf{n}^2$$

- Total execution time $= (n-1) + n^2 = \textbf{O(n}^2\textbf{)}$

# Best Case Analysis

- $L_2$, $L_8$ to $L_{10}$ executes one time each for every iteration.

$$1 + 1 + 1 + 1 + \cdots + n - 1 = \textbf{n-1}$$

- $L_4$ to $L_6$ executes in the sequence shown below for all iterations combined.

$$(n-1) + (n-2) + (n-3) + (n-4) + \cdots$$
$$+(n-(n-3)) + (n-(n-2)) + (n-(n-1))$$

$$= (n-1) + (n-2) + (n-3) + (n-4) + \cdots + 3 + 2 + 1$$
$$= \frac{n \times (n-1)}{2} = \frac{n^2}{2} = \textbf{n}^2$$

- Total execution time $= (n-1) + n^2 = \textbf{O(n}^2\textbf{)}$

# Average Case Analysis

- $L_2$, $L_8$ to $L_{10}$ executes one time each for every iteration.

$$1 + 1 + 1 + 1 + \cdots + n - 1 = \textbf{n-1}$$

- $L_4$ to $L_6$ executes in the sequence shown below for all iterations combined.

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

$$= \frac{n \times (n-1)}{4} = \frac{n^2}{4} = \textbf{n}^2$$

- Total execution time $= (n-1) + n^2 = \textbf{O(n}^2\textbf{)}$

# Another Sorting Algorithm

**Next: Quick Sort Algorithm**

# Reading Assignment

**Sedgewick 2.3 Selection Sort**

# Questions?

**Please ask if there are any Questions!**