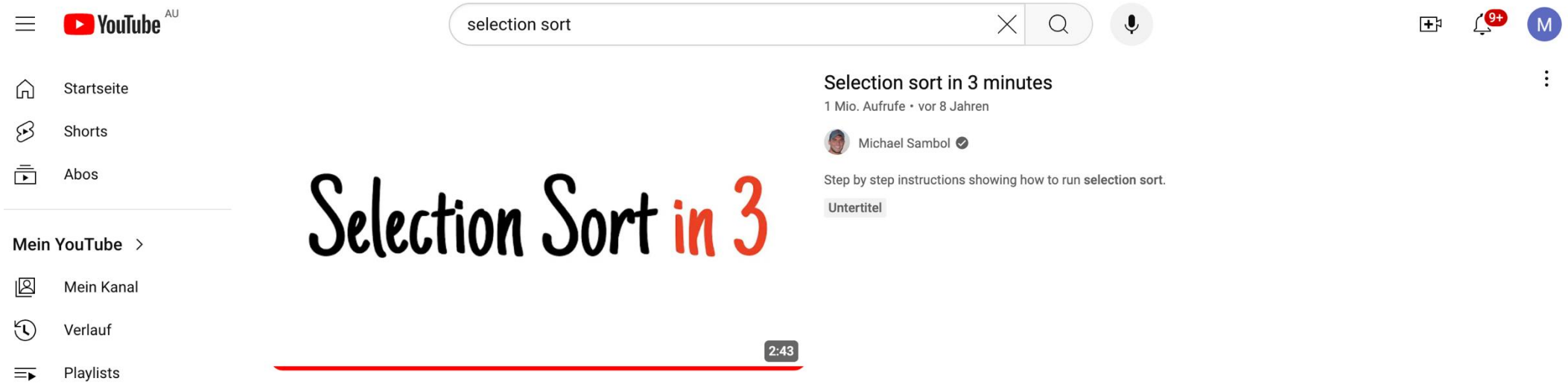
Hello Class!

If you aren't familiar with **selection sort** or would like a quick refresher, please watch the following video:

https://www.youtube.com/watch?v=g-PGLbMth_g



Tips for the MST

- Do **all past tests** (if you can find some) — they repeat patterns!
- Do the **sample test** (this year's) **twice**
- practice coding on paper (without using the lecture notes, Ed, code editors, LLMs, …)

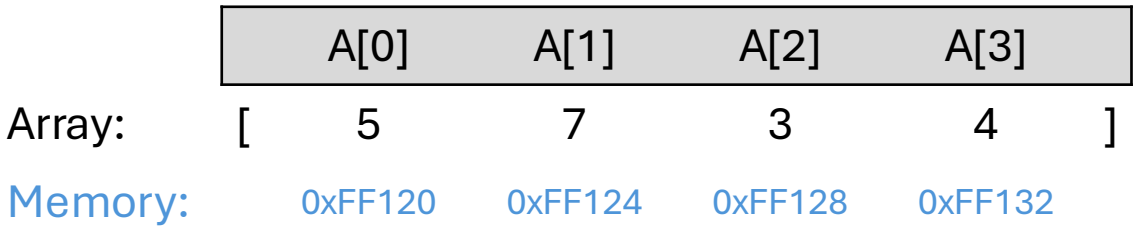
⚠️ Common coding mistakes:

- **Integer division:** `5/2` will return 2, **not** 2.5!
  Use `5.0/2.0`, or cast: `(double)5/2`

- **Loop variable not declared**
  - `for (int i=0; i<n; i++)` ✅ don't forget `int`

- **For non-void functions:**
  - Make sure you **return something** in **every possible path**!

Arrays



- an array is a collection of same-type variables
- in C, an array is a sequential block of memory → you must define the size at creation and can't change it later on
- **you cannot use negative indices! it will not give you the result you expect**

Initialising arrays

type of the array

shape of the array

access the i-th entry of the array

```c
/*  Initialise directy */

int A[3] = {1, 2, 3};

int A[5] = {1, 2};  // A[2] ... A[4] automatically set to 0

int A[2][3] = {     // 2d array
    {1, 2, 3},
    {4, 5, 6}
}


/* Fill up with a loop */

int A[3];

for (int i = 0; i < 3; i++) {
    A[i] = i + 1;
}

/* THIS DOES NOT WORK */

int A[3];
A = {1, 2, 3};
```

Arrays are **passed as pointers**

```
4   void print_array(int A[], int n) {
5       for (int i = 0; i < n; i++) {
6           printf("%d ", A[i]);
7       }
8   }
```

- When you pass an array to a function like this, you're passing a **pointer to the first element**
- Changes to A[i] inside the function **affect** the original array

There is **no built-in .shape or .length** property like in Python.

BE CAREFUL WITH THE **SIZEOF()** FUNCTION!

- If the array is defined **inside a function** e.g. as `int A[5]`, then **sizeof(A) does return** `5 * sizeof(int)`
- If you want the number of elements in an array, you can use

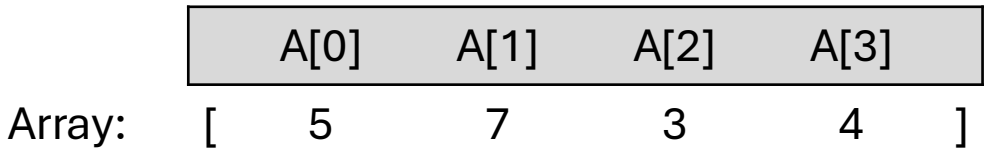
```
int count = sizeof(a) / sizeof(a[0])
```

divide by the size of int

- This works **only** if the array is in scope as a full array (i.e. was defined inside the function), otherwise it will return the size of the pointer

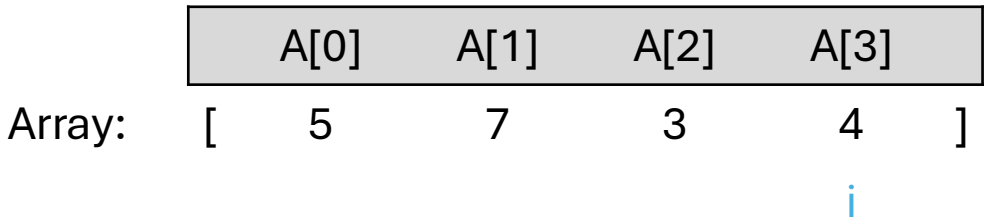
Workshop Excercises (Ed)

Selection Sort



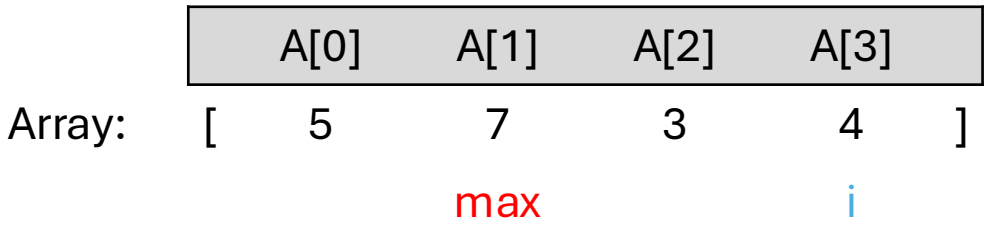
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



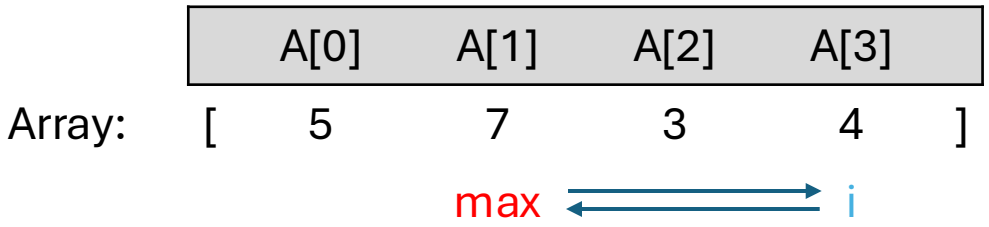
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



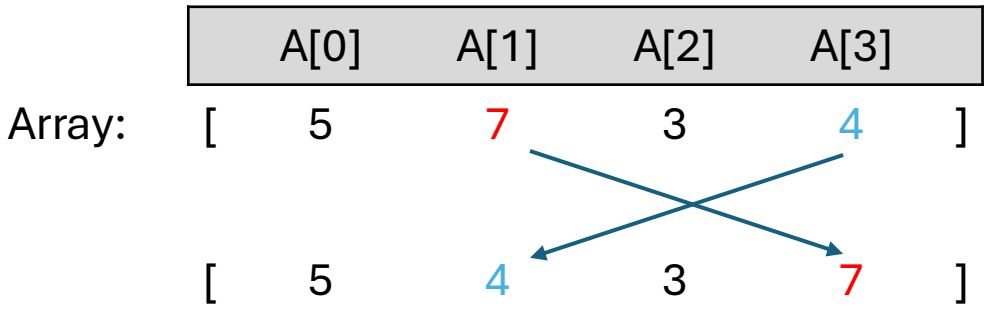
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



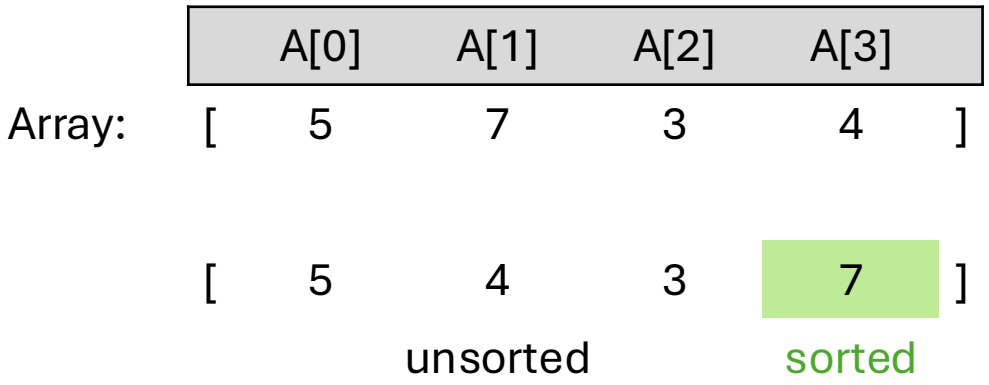
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



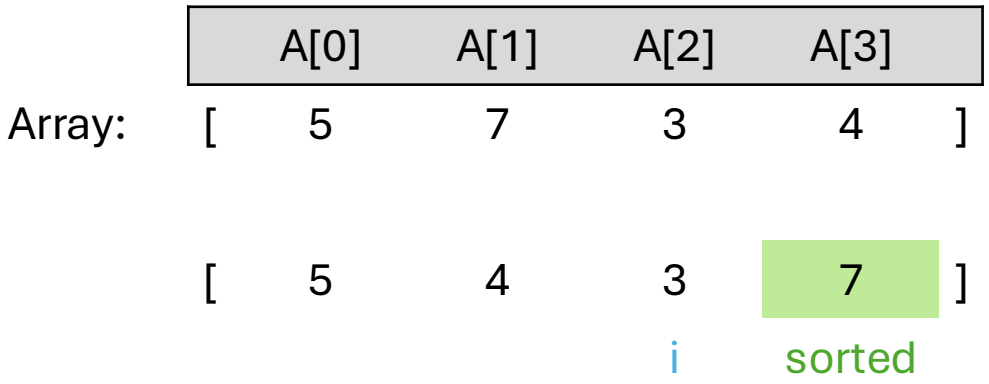
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort

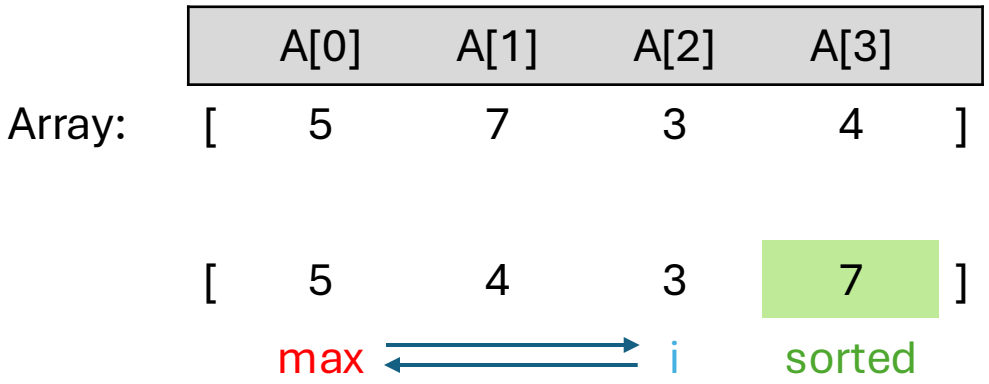


- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort

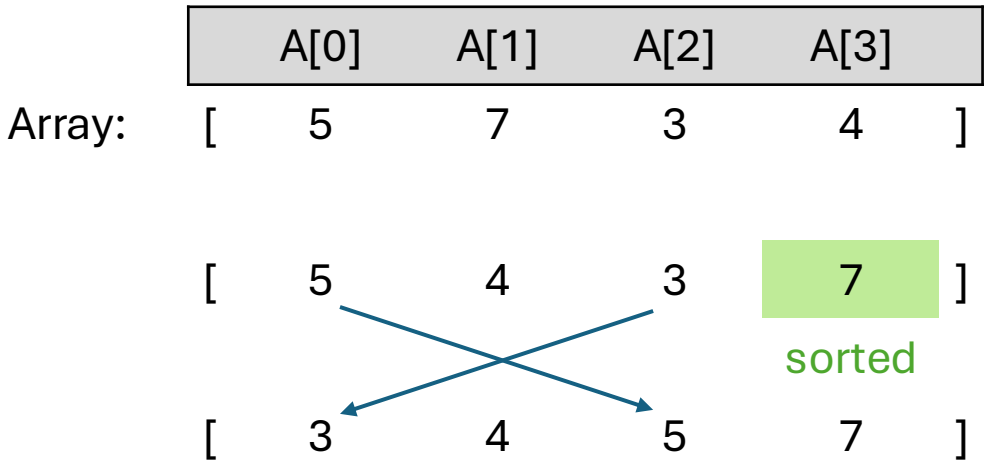
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



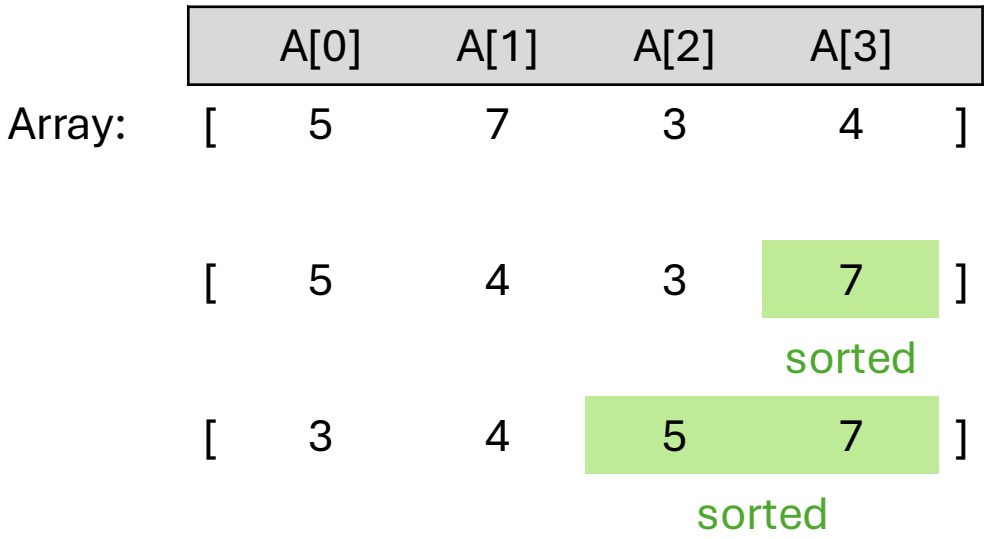
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



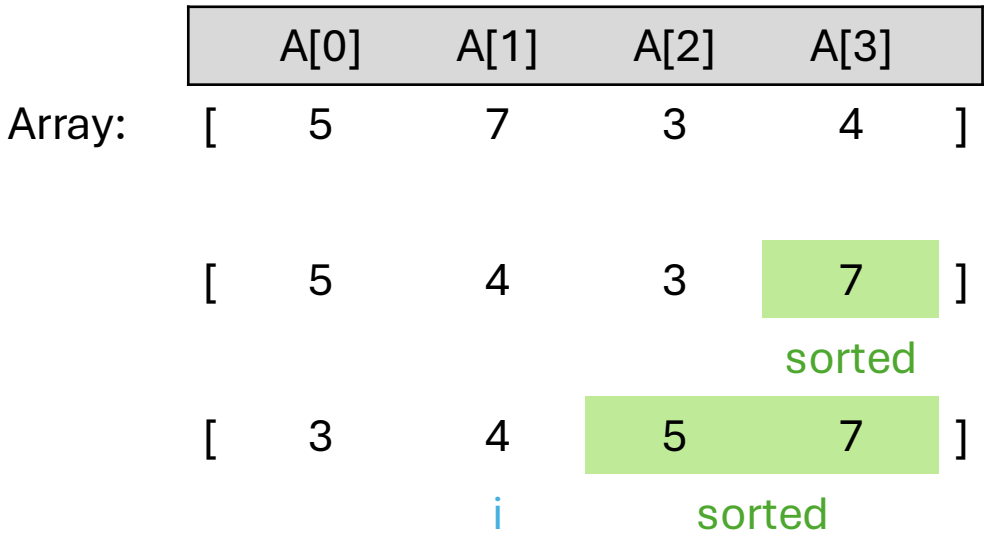
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



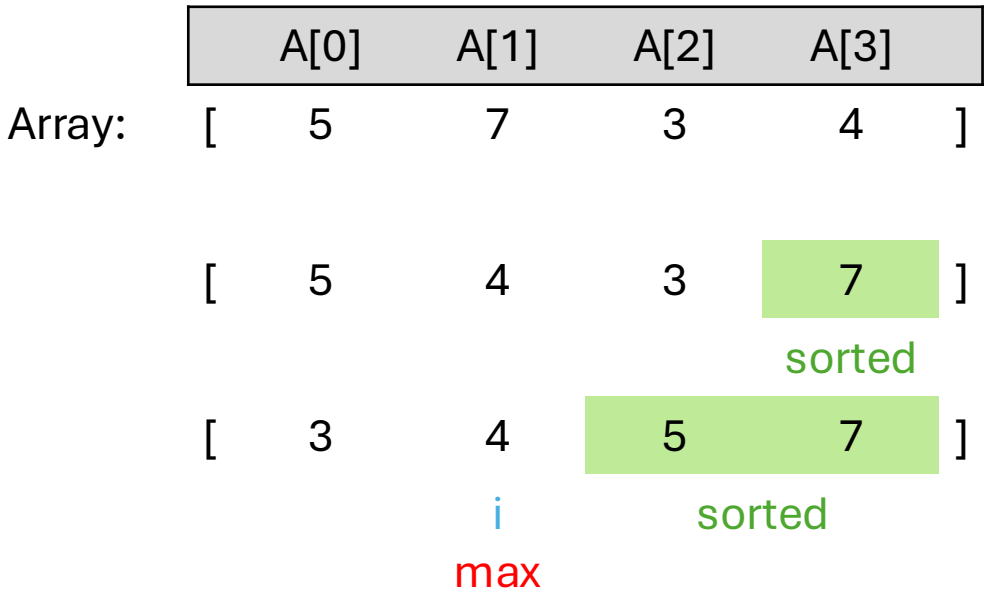
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



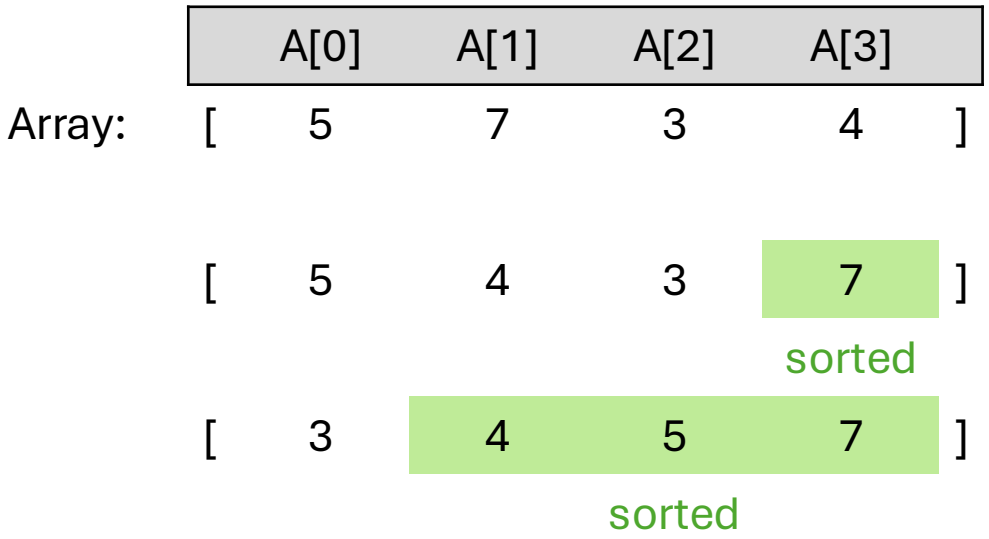
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



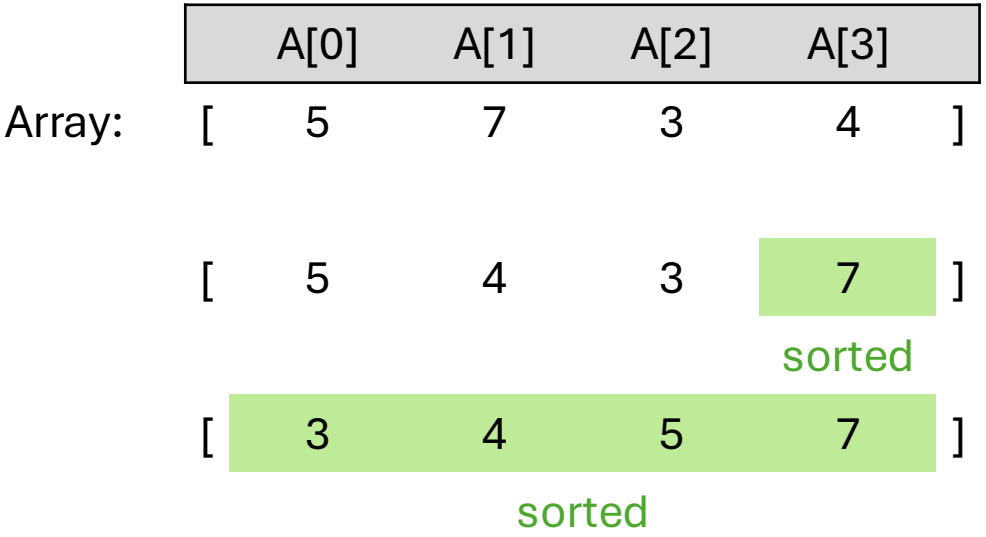
- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section

Selection Sort



- go through unsorted section of the array
- find the largest value
- swap this value into the sorted section