

## Course: Data Structures (CSE CS203A)

### Assignment II: Array Selection Sort

**Due Date:** October 7, 2025, 5:00 PM (Room R1102)

**Important Notice:** You must print this assignment (Student Worksheet Companion) and **write your answers by hand with a pen.**

#### Goal

Practice representing data in arrays, implement the selection sort algorithm step-by-step, and analyze algorithmic trade-offs including number of comparisons, swaps, and visualization of the sorting process.

#### Given Data

Input integers: 22, 90, 95, 100, 71, 19, 5, 70

#### Background

**Array:** A contiguous block of memory that stores elements of the same data type. Arrays provide  $O(1)$  random access to elements by index, but insertion and deletion operations in the middle are costly  $O(n)$  due to the need to shift elements.

**Selection Sort Algorithm:** An in-place comparison-based sorting algorithm that divides the array into sorted and unsorted portions. For each position  $i$ , it finds the minimum element in the remaining unsorted portion and swaps it into position  $i$ .

#### Selection Sort Algorithm

**Basic Idea:** For each position  $i$  from 0 to  $n-2$ , find the minimum element in the unsorted suffix (from position  $i+1$  to  $n-1$ ) and swap it into position  $i$ .

#### Pseudocode:

```
SELECTION-SORT(A[0..n-1]):
  for i = 0 to n-2:
    min_idx = i
    for j = i+1 to n-1:
      if A[j] < A[min_idx]:
        min_idx = j
    if min_idx != i:
      swap A[i] with A[min_idx]
```

**Time Complexity:**  $O(n^2)$  in all cases (best, average, worst)

**Space Complexity:**  $O(1)$  - in-place sorting algorithm

#### Tasks

Approach: Represent the given integers in an array data structure and apply the selection sort algorithm step by step.

## **Required Answer Components**

**A1:** Draw the visual representation of the array structure

**A2:** Populate the array with the given integers and add appropriate annotations (indices, values)

**A3:** Demonstrate the first three steps of the selection sort algorithm with detailed traces

## **Deliverables**

Submit a single handwritten paper containing:

- Array structure drawings (A1)
- Populated array with given data (A2)
- Step-by-step traces of the first 3 selection sort iterations (A3)

## **Evaluation Criteria (Total: 100 points)**

- Array representation & data population (A1-A2): 40 points
- Selection sort step-by-step execution (A3): 40 points
- Presentation quality & clarity: 20 points

## Student Worksheet Companion

Student Name:

### A1. Array Representation Drawing

Instructions: Draw a visual representation of an array structure that can hold 8 integers. Include:

Array cells/boxes

Index labels (0 through 7)

## Clear indication of array bounds

Complete your populated array here:

Array	[	]	[	]	[	]	[	]	[	]	[	]	[	]	[	]
Index	0		1		2		3		4		5		6		7	

## A2. Populate Array with Given Integers

Instructions: Fill the array structure you drew in A1 with the given integers: 22, 90, 95, 100, 71, 19, 5, 70

Add the following annotations:

Array name (e.g., "Array A")

Index numbers below each cell

Value labels above or inside each cell

Array A

Array	[ 22 ]	[ 90 ]	[ 95 ]	[ 100 ]	[ 71 ]	[ 19 ]	[ 5 ]	[ 70 ]
Index	0	1	2	3	4	5	6	7

### A3. Selection Sort – First Three Steps

Instructions: Show the detailed execution of the first three iterations of selection sort. For each step, track the array state, identify the minimum element, record any swaps performed, and show the resulting array.

Step1 ( $i = 0$ ):

Array before step (with indices)
<div> <div>[ 22 ]</div> <div>[ 90 ]</div> <div>[ 95 ]</div> <div>[ 100 ]</div> <div>[ 71 ]</div> <div>[ 19 ]</div> <div>[ 5 ]</div> <div>[ 70 ]</div> </div> <div> <div>0</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> </div>
Searching range: indices 0 to 7
Minimum element found: Value = <u>5</u> , Index = <u>6</u>

Swap performed: Index 0 <-> Index <u>  6  </u>							
(Circle YES or NO): <b>YES</b> / <del>NO</del>							
Array after step (with indices)							
[ 5 ]	[ 90 ]	[ 95 ]	[ 100 ]	[ 71 ]	[ 19 ]	[ 22 ]	[ 70 ]
0	1	2	3	4	5	6	7

Step2 (i = 1):

Array before step (with indices)							
[ 5 ]	[ 90 ]	[ 95 ]	[ 100 ]	[ 71 ]	[ 19 ]	[ 22 ]	[ 70 ]
0	1	2	3	4	5	6	7
Searching range: indices 1 to 7							
Minimum element found: Value = <u>  19  </u> , Index = <u>  5  </u>							
Swap performed: Index 1 <-> Index <u>  5  </u>							
(Circle YES or NO): <b>YES</b> / <del>NO</del>							
Array after step (with indices)							
[ 5 ]	[ 19 ]	[ 95 ]	[ 100 ]	[ 71 ]	[ 90 ]	[ 22 ]	[ 70 ]
0	1	2	3	4	5	6	7

s

Step3 (i = 2):

Array before step (with indices)							
[ 5 ]	[ 19 ]	[ 95 ]	[ 100 ]	[ 71 ]	[ 90 ]	[ 22 ]	[ 70 ]
0	1	2	3	4	5	6	7
Searching range: indices 2 to 7							
Minimum element found: Value = <u>  22  </u> , Index = <u>  6  </u>							
Swap performed: Index 2 <-> Index <u>  6  </u>							
(Circle YES or NO): <b>YES</b> / <del>NO</del>							
Array after step (with indices)							
[ 5 ]	[ 19 ]	[ 22 ]	[ 100 ]	[ 71 ]	[ 90 ]	[ 95 ]	[ 70 ]
0	1	2	3	4	5	6	7