

Worksheet – Algorithm

Module: Algorithm

Duration: 90 minutes

Level: Beginner

Target group: Deaf youth learning about data and coding

Part 1: Warm-up (Understanding the Basics)

1. Sum of two Digits

Sum of Two Digits Problem

Compute the sum of two single digit numbers.

Input: Two single digit numbers.

Output: The sum of these numbers.

Example: $2 + 3 = 5$

Implementing an Algorithm

- Pseudocode
- C++
- Java
- Python

2. Maximum Pairwise Product

Maximum Pairwise Product Problem

Find the maximum product of two distinct numbers in a sequence of non-negative integers.

Input: A sequence of non-negative integers.

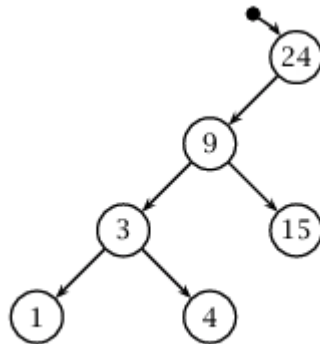
Output: The maximum value that can be obtained by multiplying two different elements from the sequence.

Example:

	5	6	2	7	4
5		30	10	35	20
6	30		12	42	24
2	10	12		7	4
7	35	42	14		28
4	20	24	8	28	

Activity: “Binary search tree” Activity

Consider the following binary search tree (BST).



Question 1: List all the possible insertion orders (i.e., permutations) of the keys that could have produced this BST.

Question 2: Draw the same BST after the insertion of keys: 6, 45, 32, 98, 55, and 69, in this order.

Question 3: Draw the BST resulting from the deletion of keys 9 and 45 from the BST resulting from question 2.

Question 4: Write at least three insertion orders (permutations) of the keys remaining in the BST after question 3 that would produce a balanced tree (i.e., a minimum-height tree).

Part 3 — Practice: “Insertion-Sort”

Write the pseudocode of the in-place insertion-sort algorithm, and illustrate its execution on the array

$$A = (7, 17, 89, 74, 21, 7, 43, 9, 26, 10)$$

Do that by writing the content of the array at each main (outer) iteration of the algorithm.

Part 4 — Practice: “max-heap”

Consider the following max-heap

$H = 37, 12, 30, 10, 3, 9, 20, 3, 7, 1, 1, 7, 5$

Write the exact output of the following `Extract-All` algorithm run on H

`Extract-All (H)`

```
1 while H.heap-size > 0
2   Heap-Extract-Max (H)
3   for i = 1 to H.heap-size
4     output H[i]
5   output “.” end-of-line
```

`Heap-Extract-Max (H)`

```
1 if H.heap-size > 0
2   k = H[1]
3   H[1] = H[H.heap-size]
4   H.heap-size = H.heap-size - 1
5   Max-Heapify (H)
6 return k
```

Part 5 — Quick Quiz

Tick (✓) the correct answer in each question.

1. What is an algorithm?
 - A. A computer brand
 - B. Step-by-step instructions to solve a problem
 - C. A type of data
 - D. A password

2. Which of these is an everyday example of an algorithm?
 - A. A cooking recipe
 - B. A football
 - C. A music video
 - D. A picture

3. What do we call the information we put into an algorithm?
 - A. Output
 - B. Input
 - C. Error
 - D. Flowchart

4. What happens if an algorithm has a mistake (bug)?
 - A. It works faster
 - B. It stops or gives wrong results
 - C. It becomes safer
 - D. It disappears

5. What is the correct order of an algorithm?
 - A. Output → Process → Input
 - B. Process → Input → Output

C. Input → Process → Output

D. Input → Output → Process

Part 6 — Reflection

Answer with short phrases or sentences.

1. What did you learn about **algorithm** today?

2. What was **difficult** for you?
