

Assignment 1

Question 1: Proving Algorithm Correctness (3 marks)

Consider the **Linear Search** algorithm:

```
LinearSearch(A, n, x):  
  for i = 0 to n-1:  
    if A[i] == x:  
      return i  
  return -1
```

- Define an appropriate **loop invariant** for the above algorithm.
 - Prove the correctness of the algorithm using the **Initialization, Maintenance, and Termination** framework.
-

Question 2: Complexity Analysis (6 marks)

Given the following functions:

- $f_1(n) = 3n^2 + 10n + 5$
- $f_2(n) = 2n + n^3$
- $f_3(n) = n \cdot \log n + 20n$

For each function:

- Determine its **Big-O** complexity.
 - Determine its **Big-Ω** complexity.
 - Determine its **Big-Θ** complexity.
- Provide step-by-step reasoning for each answer.
-

Question 3: Recurrence Relations (6 marks)

Selection Sort is defined as:

- Find the minimum element in the array.
- Swap it with the first element.
- Recursively sort the remaining subarray.

- a) Write the **recurrence relation** for the running time of Selection Sort.
 - b) Solve the recurrence using the **recursion tree method**.
 - c) Verify the solution using the **substitution method**.
-

Question 4: Programming Exercise – Divide & Conquer (5 marks)

Kth Largest Element in an Array

Given an integer array `nums` and an integer `k`, return the **kth largest element** in the array.

Example:

Input: `nums = [3,2,1,5,6,4]`, `k = 2`

Output: 5

Requirements:

- a) Solve this problem using a **Divide and Conquer** approach.
- b) Analyze the **time complexity** and **space complexity** of your solution.

Instruction: Implement the solution in Python, and you must **NOT** use sorting to solve the problem.