**TIME COMPLEXITY**

**DATA STRUCTURE AND ALGORITHM**

**public class BigOPractice {**

**public static int sumArray(int[] arr) {**

**int sum = 0;**

**for (int num : arr) {**

**sum += num;**

**}**

**return sum;**

**}**

**public static void printAllPairs(int[] arr) {**

**for (int i : arr) {**

**for (int j : arr) {**

**System.out.println("(" + i + ", " + j + ")");**

**}**

**}**

**}**

**public static int binarySearch(int[] arr, int target) {**

**int low = 0, high = arr.length - 1;**

**while (low <= high) {**

**int mid = (low + high) / 2;**

**if (arr[mid] == target) return mid;**

**else if (arr[mid] < target) low = mid + 1;**

**else high = mid - 1;**

**}**

**return -1;**

**}**

**public static void process(int[] arr) {**

**for (int i = 0; i < arr.length; i++) {**

**int j = 1;**

**while (j < arr.length) {**

**System.out.println("Processing: " + arr[i] + ", step = " + j);**

**j \*= 2;**

**}**

**}**

**}**

**public static int getFirst(int[] arr) {**

**return arr.length > 0 ? arr[0] : -1;**

**}**

**public static int factorial(int n) {**

**if (n == 0) return 1;**

**return n \* factorial(n - 1);**

**}**

**public static int fibonacci(int n) {**

**if (n <= 1) return n;**

**return fibonacci(n - 1) + fibonacci(n - 2);**

**}**

**public static void main(String[] args) {**

**int[] nums = {1, 2, 3, 4, 5};**

**int[] smallNums = {1, 2, 3};**

**System.out.println("Sum = " + sumArray(nums));**

**System.out.println("First Element: " + getFirst(nums));**

**System.out.println("\nAll Pairs:");**

**printAllPairs(smallNums);**

**System.out.println("\nBinary Search Result: " + binarySearch(nums, 3));**

**System.out.println("\nProcessing with Logarithmic Inner Loop:");**

**process(nums);**

**int factNum = 5;**

**System.out.println("\nFactorial of " + factNum + " is: " + factorial(factNum));**

**int fibNum = 6;**

**System.out.println("Fibonacci of " + fibNum + " is: " + fibonacci(fibNum));**

**}**

**}**