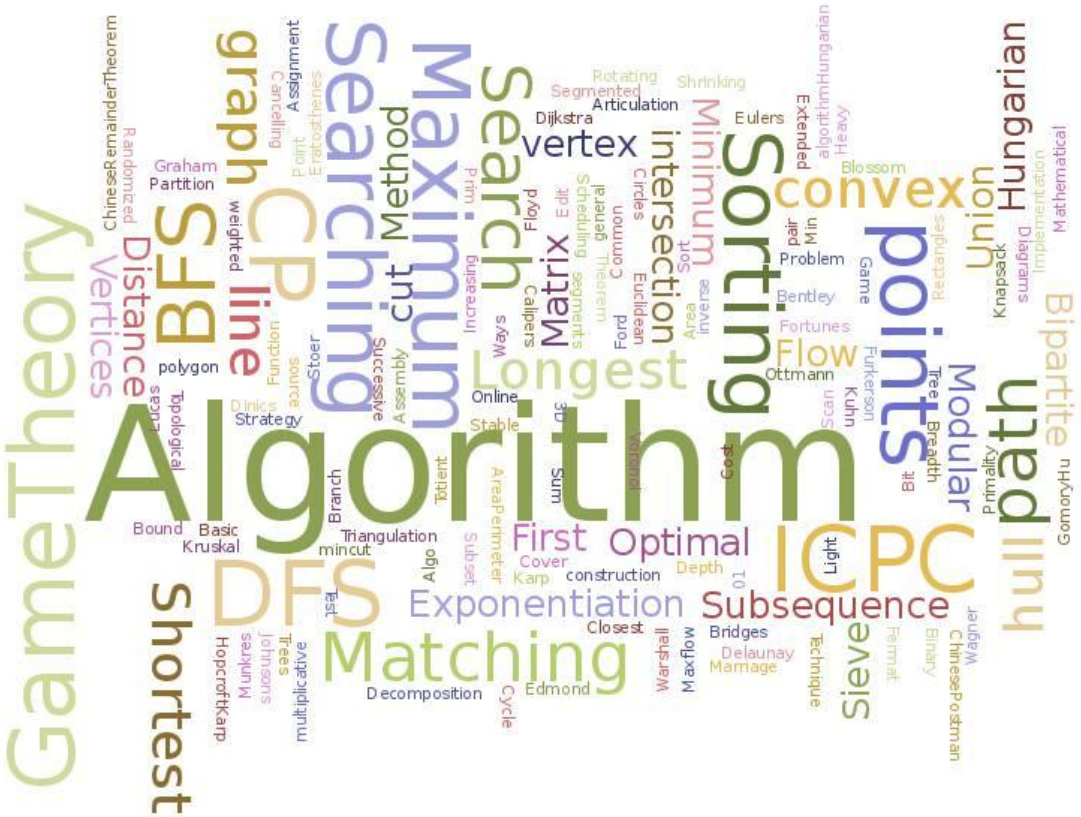
AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING

Design and Analysis of Algorithms Lab 1 Report

(Course Code: 23CSE211)



Name: NAGARAMPALLI SARVAN KUMAR

Roll Number: CH.SC.U4CSE24130

Branch/ Section: Computer Science and Engineering / CSE – B

College: Amrita Vishwa Vidyapeetham Chennai Campus

Academic Year: 2025 – 2026

Programs and Their Space Complexities Date: 27-11-2025

1. Write a program to find the sum of first *n* natural numbers using a function.

2. Write a program to find the sum of squares of first *n* natural numbers.

3. Write a program to find the sum of cubes of first *n* natural numbers.

4. Write a program to find factorial of a number using a recursive function.

5. Write a program to find the transpose of a 3×3 matrix.

6. Write a program to print the Fibonacci series using recursion.

Solutions:

Write a program to find the sum of first *n* natural numbers using a function.

```c
#include <stdio.h>

int sumOfNums(int n){
  return ((n*(n+1))/2);
}

int main(){
  int  n;
  scanf("%d", &n);
  int sum = sumOfNums(n);
  printf("The sum is: %d\n", sum);
}
```

Justification: Uses the formula n(n+1)/2, which calculates the total directly.
Space Complexity: O(1)

Write a program to find the sum of squares of first *n* natural numbers.

```c
#include <stdio.h>

int sumOfSquaresNums(int n){
  return ((n*(n+1)*(2*n+1))/6);
}

int main(){
  int  n;
  scanf("%d", &n);
  int sum = sumOfSquaresNums(n);
  printf("The sum is: %d\n", sum);
}
```

Justification: Uses the formula n(n+1)(2n+1)/6, computes the sum directly.

Space Complexity: O(1)

Write a program to find the sum of cubes of first *n* natural numbers.

```c
#include <stdio.h>

int sumOfCubeNums(int n){
  return ((n*n*(n+1)*(n+1))/4);
}

int main(){
  int  n;
  scanf("%d", &n);
  int sum = sumOfCubeNums(n);
  printf("The sum is: %d\n", sum);
}
```

Justification: Uses the formula $[(n(n+1))/2]^2$, which calculates the total directly.

Space Complexity: O(1)

Write a program to find factorial of a number using a recursive function.

```c
#include <stdio.h>

int factorial(int n){
    if(n ==0 || n == 1){
      return 1;
    }
    else{
      return n*factorial(n-1);
    }
}

int main(){
  int n;
  scanf("%d", &n);
  int res = factorial(n);
  printf("The factorial is: %d\n", res);
}
```

Justification: Uses recursive calls that reduce n step-by-step until the base case is reached.
Space Complexity: O(n)

Write a program to find the transpose of a 3×3 matrix.

```c
#include <stdio.h>

int main(){
  int arr[3][3];
  for(int i = 0; i<3; i++){
    for(int j = 0; j<3; j++){
      scanf("%d", &arr[j][i]);
      }
  }

  for(int i = 0; i<3; i++){
    for(int j = 0; j<3; j++){
      printf("%d ", arr[i][j]);
      }
      printf("\n");
  }
}
```

Justification: Uses a fixed 3×3 array and stores each element directly in its transposed position during input.

Space Complexity: O(1)

Write a program to print the Fibonacci series using recursion.

```c
#include <stdio.h>

int fibonacci(int n){
  if(n == 1){
    return 0;
  } else if (n == 2){
    return 1;
  }
  else{
    return fibonacci(n-1) + fibonacci(n-2);
  }
}

int main(){
  int n;
  scanf("%d", &n);
  for(int i = 1; i⩽n; i++){
    printf("%d ", fibonacci(i));
  }
}
```

Justification: Uses recursive calls that expand until the base cases are reached.
Space Complexity: O(n)