1. Find the M-th maximum number and Nth minimum number in an array and then find the sum and difference of it.

Test cases: output –

* 1. {16, 16, 16 16, 16}, M = 0, N = 1 (illegal input)
  2. {0, 0, 0, 0}, M = 1, N = 2 0
  3. {-12, -78, -35, -42, -85}, M = 3 , N = 3 -7
  4. {15, 19, 34, 56, 12}, M = 6 , N = -3 (illegal input)
  5. {85, 45, 65, 75, 95}, M = 5 , N = 2 -20

1. Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1.integer target. Write a program to search a number in a list using binary search and estimate time complexity

Test cases:

Input : ( 45, 4, 23, -11, 20, 5, 10, 50) Key element 5 Output Found in the position 2

Input : ( 8,-2, 11, 8, 6, 3 10,0) Key element 2 Output Not found

1. Write a program to find the reverse of a given number. Find and write the time complexity

Input / Output

1234 - 4321

67894 - 49876

45a34 - Illegal input

1. Write a program to compute Binomial coefficient for n=8, k=8 using dynamic programming

Using condition such as

I nCk =1 if k=0 or n=k

II nCk – (n-1)Ck-1 + (n-1)Ck for n>k>0

1. Write a program to perform sum of subsets problem using backtracking and find the time complexity.

Input / Output

Input : Set (s) = (6, 2,8,1,5) sum is 9 Set (s) = (6, -4, 7, -1, 5, 2,8,1,) sum is 10

Output : Subset is (6,2,1) (2,8,1) Subset is (6,-4, 8) (2,8)

1. Write a program to check the given number is Armstrong or not.

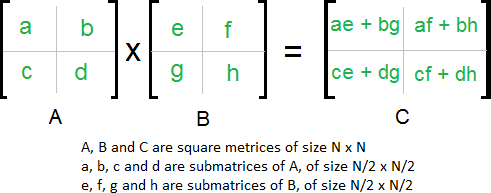
The k-digit number N is an Armstrong number if and only if the k-th power of each digit sums to N.

Given a positive integer N, return true if and only if it is an Armstrong number. Input : 153 Input : 419

Output : True Output : False

1. Write a program to perform Strassen’s Matrix Multiplication for the 2\*2 matrix elements. Find its time complexity.

Example:



1. Write a program to find the Factorial of a number using recursive method and write its time complexity.
2. Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1. You must write an algorithm with O(log n) runtime complexity.
3. Write a program to find the GCD of two numbers. Find time complexity if recursion is

used Perform the test cases for the given set of no’s

|  |  |  |  |
| --- | --- | --- | --- |
| A. | (36,48) |  | 2 |
| B. | (156, 90) | 6 |  |

C. (-56,88) Illegal input

1. Find Max and Min value in the list using divide and conquer find its time complexity.

Testing Condition – Count the number of times in Comparison to find Min\_Max value in a list n for the given set of elements.

* + (23,45,6,8,-9,44,7,8) Min val = -9, Max Value = 45
  + (8,-5,7,2,6,0,1,9) Min val = -5, Max Value = 9
  + (45, y, 9, 8,4, 7,11, 22,16) Illegal input

1. Generate a program for Pascal triangle. Estimate the time complexity for the row=5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 1 |  |  |
|  | 1 | 1 |  |  |
|  | 1 | 2 | 1 |  |
| 1 | 3 | 3 | 1 |  |
| 1 | 4 | 6 | 4 | 1 |
|  |  |  |  |  |

1. Write a program to find the sum of digits. You are given a **0- indexed** array nums consisting of **positive** integers. You can choose two indices i and j, such that i != j, and the sum of digits of the number nums[i] is equal to that of nums[j]. Return *the* **maximum** *value of* nums[i] + nums[j] *that you can obtain over all possible indices* i *and* j *that satisfy the conditions.*
2. Consider a two integer arrays nums1 and nums2, sorted in non-increasing order and two integers m and n, representing the number of elements in nums1 and nums2 respectively. Write a program to Merge them into a single array using Merge Sort. Derive time complexity of merge sort

.Input Set[], A = (3,8,1,9) Set[], B = (4,-2, 0,7) Output A \* B = (-2,0,1,3,4,7,9)

1. Write a program to find all pairs shortest path using Floyd's technique and to estimate its time complexity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A | 0 | 8 | 7 | 8 |
| B | 9 | 0 | 11 | 12 |
| C | 10 | 9 | 0 | 11 |
| D | 8 | 10 | 11 | 0 |

1. Write a program to perform linear search and estimate time complexity. Compute the amount of time for completion.

Input/ Output series

|  |  |  |
| --- | --- | --- |
| A = (56,89,7,13,75, 23, 8, 12) | Key element 75 | Element found in position 4 |
| B = (89,45 -23,45,0, 44, 2) | Key element 0 | Element found in position 5 |
| C = (45,67,56,A,34,-2,100) | Key element 90 | Not Found |