**S.Tejeswar – AI & DS (22AD104)**

**22/11/2024 and 23/11/2024**

**1 . Prime numers in a given range:**

**C++ program:**

class Solution {

public:

int isPrime(int num){

if(num == 2){

return 1;

}

if(num < 3 || num % 2 == 0) return 0;

for (int i = 3; i <= sqrt(num); i += 2){

if(num % i == 0){

// cout << "is prime" << i;

return 0;

}

}

return 1;

}

vector<int> primeRange(int M, int N) {

vector<int> answer;

for(int i = M; i < N+1; i++){

if(isPrime(i) && i > 1){

// cout << i<< ' ';

answer.push\_back(i);

}

}

return answer;

}

};

**Time Complexity: O(N.sqrt(N))**

**Space Complexity: O(1)**

**2. Armstrong numbers between two intervals**

**C++ Program:**

vector<int> getArmstriongNumbers(int m, int n){

int num;

vector<int> result;

for(num = m; num <= n + 1; num++) {

int temp = num;

int digits = 0;

while(temp){

digits++;

temp = temp / 10;

}

// cout << "digits" << digits << '\n';

temp = num;

int armstrongAnswer = 0;

while(temp){

armstrongAnswer += pow((temp % 10), digits);

temp = temp / 10;

}

if(armstrongAnswer == num){

result.push\_back(num);

}

}

return result;

}

int main()

{

vector<int> result = getArmstriongNumbers(1, 500);

for(auto num : result){

cout << num << ' ';

}

}

Time Complexity: O(NlogN)

Space Complexity: O(1)

**3. Can a number be expressed as a sum of two prime numbers?**

**C++ Program:**

int isPrime(int n){

if(n == 2) return 1;

if(n < 2 || n%2 == 0) return 0;

for(int i = 3; i <= sqrt(n); i+= 2){

if(n % i == 0){

return 0;

}

}

return 1;

}

vector<int> getPrimes(int n) {

if(n <= 2){

return {-1, -1};

}

for(int i = 2; i <= n / 2; i++){

if(isPrime(i) && isPrime(n - i)){

return {i, n - i};

}

}

return {-1, -1};

}

Time Complexity: O(N.sqrt(N))

Space Complexity: O(1)

**4. Replace all 0's with 1 in a given integer:**

C++ Program:

int replaceWith1(int n){

int answer = 0;

int rem;

while(n > 0){

rem = n % 10;

if(rem == 0){

answer = answer\* 10 + 1;

}

else{

answer = answer \* 10 + rem;

}

n = n/10;

}

int temp = 0;

while(answer > 0){

temp = temp \* 10 + (answer % 10);

answer = answer / 10;

}

return temp;

}

int main()

{

int ans = replaceWith1(123009900);

cout << ans;

}

Time Complexity: O(logN)

SpaceComplexity: O(1)

**5. Binary To Decimal**

C++ Program:

int binary\_to\_decimal(string &b) {

// Code here.

int ans = 0;

int weight = 0;

int strLen = b.size();

int i = strLen - 1;

while(i >= 0){

if(b[i] == '1'){

ans += pow(2, weight);

}

weight++;

i -= 1;

}

return ans;

}

Time Complexity: O(N) -> N is length of the binary string

Space Complexity: O(1)

**6. Decimal To Binary:**

C++ Program:

void toBinary(int N)

{

int temp = 1;

while(N > 0){

temp = temp \* 10 + (N % 2);

N = N/ 2;

}

int ans = 0;

while(temp >= 10){

ans = ans \* 10 + (temp%10);

temp = temp / 10;

}

cout<< ans;

}

Time Complexity: O(logN)

Space Complexity: O(1)

**7. Decimal To Octal**

C++ Program:

void DecimalToOctal(int N)

{

int temp = 1;

while(N > 0){

temp = temp \* 10 + (N % 8);

N = N/ 8;

}

int ans = 0;

while(temp >= 10){

ans = ans \* 10 + (temp%10);

temp = temp / 10;

}

cout<< ans;

}

Time Complexity: O(logN)

Space Complexity: O(1)

**8. Octal To Decimal**

C++ Program

void octal\_to\_decimal(int N) {

// Code here.

int ans = 0;

int weight = 0;

while(N > 0){

ans = ans + (N % 10) \* pow(8,weight );

N = N / 10;

weight++;

}

cout << ans;

}

Time Complexity: O(logN)

Space Complexity: O(1)

9. Binary to Octal Conversion