DSA Lab 4

1)

int Sum=0; //O(1)

for(int i=0; i<N; i++) //O(N)

Sum++;

for(int j=0; j<N; j++) //O(N)

Sum++;

OVERALL TIMECOMPLEXITY=O(2N)

2)

int Sum=0; //O(1)

for(int i=0; i<N; i++) //O(N)

for(int j=0; j<N; j++) //O(N^2)

for(int k=0; k<N; k++) //O(N^3)

Sum++;

for(int i=0; i<N; i++) //O(N)

for(int j=0; j<N; j++) //O(N^2)

for(int k=0; k<N; k++) //O(N^3)

Sum++;

OVERALL TIME=O(2N^3)

3)

int Sum=0;

for(int i=0; i<N; i++) //O(N)

Sum++;

for(int j=0; j<N; j++) //O(N)

Sum++;

for(int k=0; k<N; k++) //O(N)

Sum++;

for(int m=0; m<N; m++) //O(N)

Sum++;

for(int n=0; n<N; n++) ) //O(N)

Sum++;

for(int p=0; p<N; p++) //O(N)

Sum++;

OVERALL TIME=O(6N)

4)

int Sum=0; //O(1)

for(int i=0; i<N; i++) //O(N)

for(int j=0; j<i; j++) //O(N^2)

for(int k=0; k<j; k++) //O(N^3)

Sum++;

OVERALL TIME=O(N^3)

5)

int Sum=0;

for(int i=0; i<N; i+=2) //O(N)

for(int j=0; j<i; j+=2) //O(N^2)

for(int k=0; k<j; k+=2) //O(N^3)

Sum++;

OVERALL TIME=O(N^3)

6)

int Sum=0;

for(int i=1; i<N; i\*=2) //O(LOG(N))

for(int j=1; j<N; j\*=2) //O(LOG(N)^2)

Sum++;

OVERALL TIME=O((LOG(N)^2)

7)

int Sum=0;

for(int i=1; i<N; i\*=2) //O(LOG(N))

Sum++;

for(int j=1; j<N; j\*=2) //O(LOG(N))

Sum++;

OVERALL TIME=O(2LOG(N))

8)

for(int i=1; i<=N\*N; i+=2) //O(2LOG(N))

for(int j=1; j<N\*N; j\*=2) //O(2LOG(N))^2

Sum++;

OVERALL TIME=O(2LOG(N))^2

9)

for(int i=1; i<=N\*N; i+=2) //O(2LOG(N))

Sum++;

for(int j=1; j<N\*N; j\*=2) //O(2LOG(N))

Sum++;

OVERALL TIME=O(4LOG(N))

10)

for(int i=1; i<=N\*N; i\*=2) //O(2LOG(N))

for(int j=1; j<N\*N; j\*=2) //O(2LOG(N))^2

Sum++;

OVERALL TIME=O(2LOG(N))^2

11)

for(int i=1; i<=N\*N; i\*=2) //O(2LOG(N))

Sum++;

for(int j=1; j<N\*N; j\*=2) //O(2LOG(N))

Sum++;

OVERALL TIME=O(4LOG(N))

12)

int Sum=0;

for(int i=1; i<=N; i\*=2) //O(2LOG(N))

for(int j=1; j<=N; j\*=2) //O(2LOG(N))^2

for(int k=1; k<=N; k\*=2) //O(2LOG(N))^3

Sum++;

OVERALL TIME=O(2LOG(N))^3

13)

int Sum=0;

for(int i=1; i<=N; i\*=2) //O(LOG(N))

Sum++;

for(int j=1; j<=N; j\*=2) //O(LOG(N))

Sum++;

for(int k=1; k<=N; k\*=2) //O(LOG(N))

Sum++;

OVERALL TIME= O(LOG(N))

14)

int sum,i,j;

sum = 0;

for (i=1;i<n;i=i\*2) //O(LOG(N))

{

for (j=0;j<n;++j) //O((NLOG(N))

{

sum++;

}

}

OVERALL TIME=O(NLOG(N))

15)

int sum,i,j;

sum = 0;

for (i=1; i<n; i=i\*2) //O(LOG(N))

{

for (j=0; j <i ; ++j) //O((N))

{

sum++;

}

}

OVERALL TIME=O((N))

16)

int sum,i,j;

sum = 0;

for (i=1; i<n; i=i\*5) //O(LOG(N))

{

for (j=0; j<i; j+=2) //O((N))

{

sum++;

}

}

OVERALL TIME=O(LOG(N))

17)

int sum,i,j;

sum = 0;

for (i=1; i<n; i=i\*4) O(LOG(N))

{

for (j=0 ; j<n ; j+=3) O((N/3))

{

sum++;

}

}

OVERALL TIME= O(NLOG(N/3))

18)

int Sum = 0;

for(int i=1; i<=n; i+=1) //O(N)

{

Sum+=i; //O(N^2)

}

cout<<Sum<<endl;

OVERALL TIME=O(N^2)

19)

int Sum = 0;

for(int i=1; i<=n; i\*=2) //O(LOGN)

{

Sum+=i; //O(N)

}

cout<<Sum<<endl;

OVERALL TIME=O(N)

20)

int Sum = 0;

for(int i=1; i<=n; i+=1) //O(N)

{

for(int j=1; j<=i; j++) //O(N^2)

{

Sum++;

}

}

cout<<Sum<<endl;

OVERALL TIME=O(N^2)

21)

int Sum = 0;

for(int i=1; i<n; i\*=2) //O(LOG(N))

{

for(int j=1; j<=i; j++) //O(N)

{

Sum++;

}

}

cout<<Sum<<endl;

}

OVERALL TIME=O(N)

22)

{

int d;

for(d=0; d\*d<=N; d++) { } //O(SQRT(N))

return d-1;

}

bool primeNumber(int n)

{

bool isPrime = true;

int lmt = (sqrt(n)); //O(SRT(N))

for (int d=2; d <=lmt ;++d)

{

if (n%d==0)

return false;

}

return true;

}

OVERALL TIME=O(SQURT(N))

23)

int sqrt(int N)

{

int d;

for(d=0; d\*d<=N; d++){ } //O(SQR(N))

return d-1;

}

bool primeNumber(int n)

{

bool isPrime = true;

for (int d=2; d <= sqrt(n) ;++d) //O(N)

{

if (n%d==0)

return false;

}

return true;

}

OVERALL TIME=O(N)

24)

int f1(int n)

{

int K=0;

for(int j=0; j\*j<=n\*n; j++) K++;`//O(N)

return K;

}

int main()

{

int Sum = 0, n;

cin>>n;

for(int i=1; i<=f1(n); i+=1) //O(N^2)

for(int j=1; j<=i; j++) Sum++; //O(N^3)

cout<<Sum<<endl;

}

OVERALL TIME=O(N^2)

25)

int f1(int n)

{

int K=0;

for(int j=1; j\*j<=n; j\*=2) K++; //O(SQRT(N))

return K;

}

int main()

{

int Sum = 0;

int n;

cin>>n;

for(int i=1; i<=f1(n); i+=1) //O(N^2)

for(int j=1; j<=i; j++) Sum++; //O(N^3)

cout<<Sum<<endl;

}

OVERALL TIME=O(N^3)

26)

int f1(int n)

{ int K=0;

for(int j=1; j\*j<=n; j++)//O(SQRT(N))

K++;

return K\*K; //O(N)

}

int main()

{

int Sum = 0;

int n;

cin>>n;

int Terminator = f1(n);

for(int i=1; i<= Terminator; i+=1) //O(N)

{

for(int j=1; j<=i; j++)//O(N^2)

{

Sum++;

}

}

cout<<Sum<<endl;

}

OVERALL TIME=O(N^2)

27)

int f1(int n)

{ int K=0;

for(int j=0; j\*j<=n; j++) //O(SQRT(N))

K++;

return K;

}

int main()

{

int Sum = 0;

int n;

cin>>n;

int Terminator = f1(n);

for(int i=1; i<=Terminator; i+=1) //O(SQRT(N))

{

for(int j=1; j<=i; j++) //O(N)

{

Sum++;

}

}

cout<<Sum<<endl;

}

OVERALL TIME=O(N)

28)

for (i=1;i<n;i=i\*4) O(LOG(N))

{

cout << i;

for (j=0;j<n;j=j+2) O(LOG(N)N)/2

{

cout << j;

sum++

}

cout << sum;

}

OVERALL TIME= O(NLOG(N)/2)

29)

for (i=1;i<n;i=i\*4) //O(LOG(N))

{

cout << i;

for (j=0;j<i; j=j+2) //O(N)

{

cout << j;

sum++

}

cout << sum;

}

OVERALL TIME=O(N)

38)

for (int i=1; i <= n ; i = i \* 2) O(LOG(N))

{

for ( j = 1 ; j <= i ; j = j \* 2) O(log(N))

{

cout<<”\*”;

}

}

for(int i=0; i<=N; i++) O(N)

{

Sum++;

}

OVERALL TIME= O(N)

39)

for (i=0; i<n; i=i+3) O(N/3)

{

cout << i;

for (j=1; j<n; j=j\*3) O(log(N)N)

{

sum++

}

}

for (k=1;k<n;k=k\*3) O(log(N))

{

cout << j;

sum++

}

cout << sum;

OVERALL TIME= O(NLOG(N))

37)

for (i=0; i<n; i=i+3) O(N/3)

{

cout << i;

for (j=1; j<n; j=j\*3) O(NLOG(N)/3)

{

cout << j;

sum++

}

for (k=1;k<n;k=k\*3) O(NLOG(N)/3)

{

cout << j;

sum++

}

cout << sum;

}

OVERALL TIME= O(2NLOG(N)/3)

36)

for (int i=1; i <= n ; i = i \* 2) O(LOG(N))

for ( j = 1 ; j <= i ; j = j \* 2) O(LOG(N))

cout<<”\*”;

for (int i=1; i <= n ; i = i \* 2) O(LOG(N))

for ( j = 1 ; j <= i ; j = j \* 2) O(LOG(N))

cout<<”\*”;

OVERALL TIME= O(2LOG(N))

35)

for (int i=1; i <= n ; i = i \* 2) O(LOG(N))

{

for ( j = 1 ; j <= i ; j = j \* 2) O(N)

cout<<”\*”;

}

OVERALL TIME= O(N)

## **Q:3**

1)

int f1(int N)

{

int Count = 0;

for(int i = 1; i<=N ; i\*= 2) O(log(N))

for(int j=1; j<= i ; j++) O(N)

Count++;

return Count;

}

OVERALL TIME= O(N)

2)

int f2(int N)

{

int Count=0;

int C = f1(N);

for(int i=0; i<C; i++) O(N)

Count++;

return Count;

}

OVERALL TIME= O(N)

3)

int f5(int N)

{

int Count=0;

for(int i=0; i<sqrt(f1(N) \* f1(N)); i++) O(N^5/2)

Count++;

return Count;

}

OVERALL TIME= O(N^5/2)

4)

int f3(int N)

{

int Count=0;

int C = sqrt(f1(N));

for(int i=1; i<C; i\*=2) O(log(N)/2)

Count++;

return Count;

}

OVERALL TIME= O(LOG(N)/2)

5)

int f4(int N)

{

int Count=0;

for(int i=0; i<f1(N) \* f1(N); i++) O(N^3)

Count++;

return Count;

}

OVERALL TIME= O(N^3)

6)

Int Sum = 0;

int f6(int N)

{

if(N==1)

return 1;

Sum +=f1(N); O(N)

Sum +=f2(N); O(N)

Sum +=f3(N); O(SQRT(N))

Sum +=f4(N); O(N^2)

Sum +=f5(N); O(N^5/2)

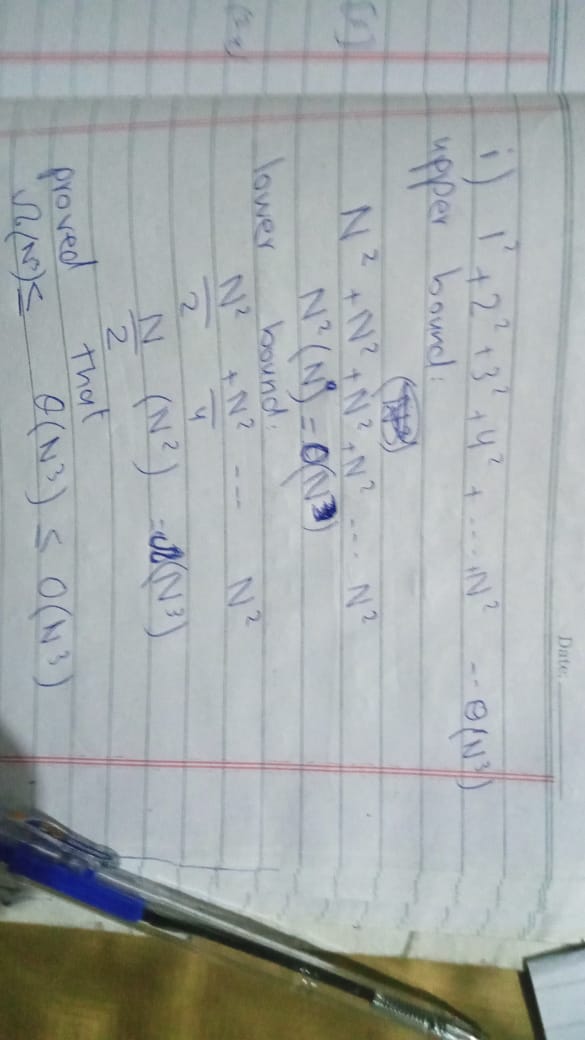
return Sum;

}

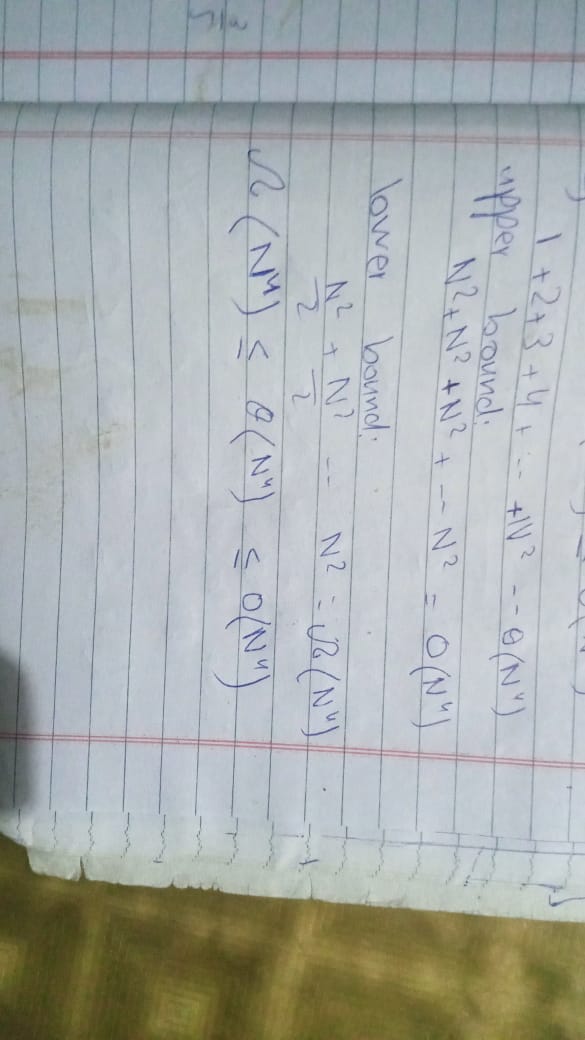
OVERALL TIME= O(N^5/2)

## Q:1

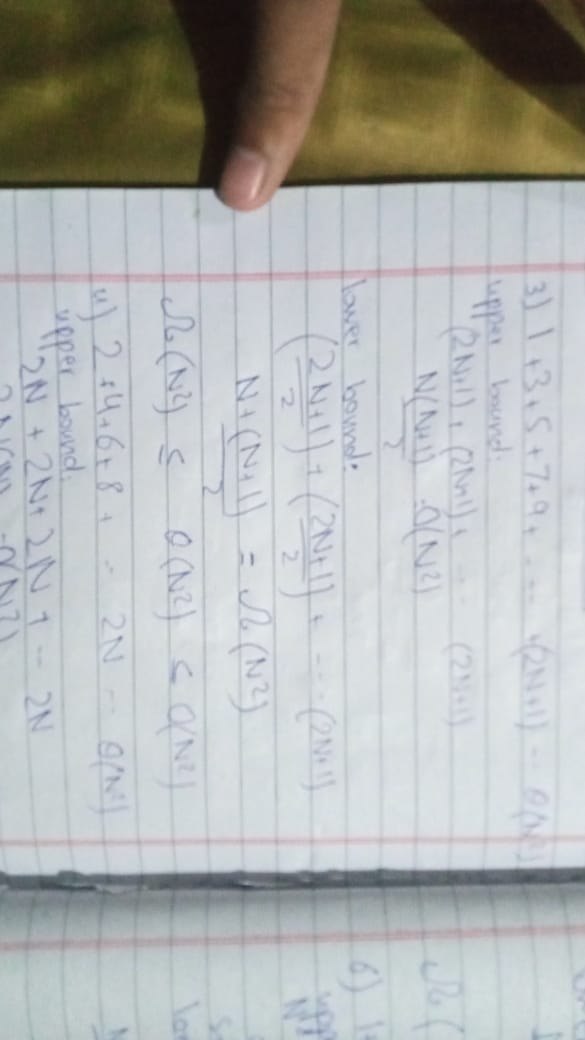
1)



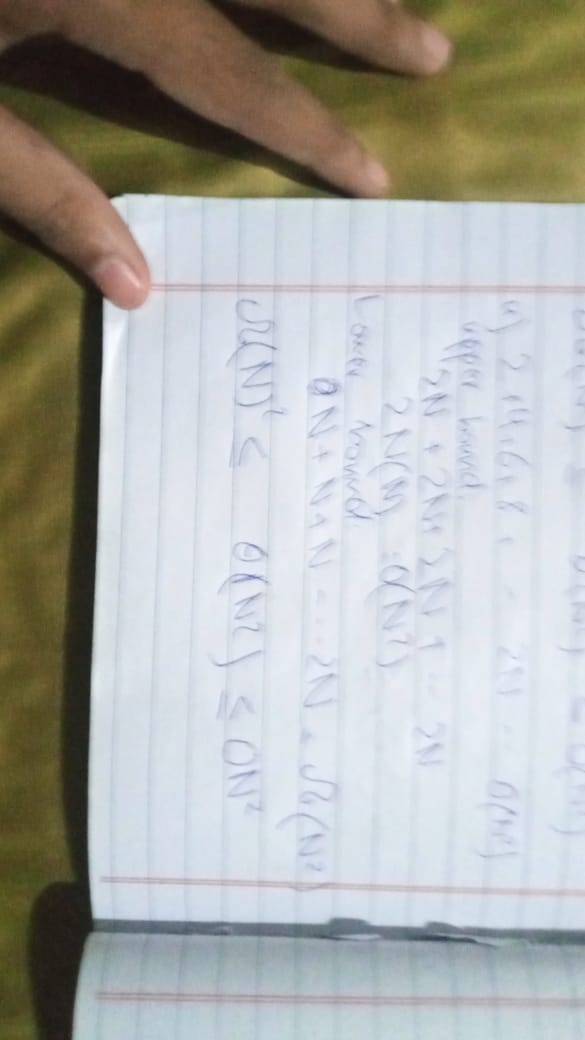
2)



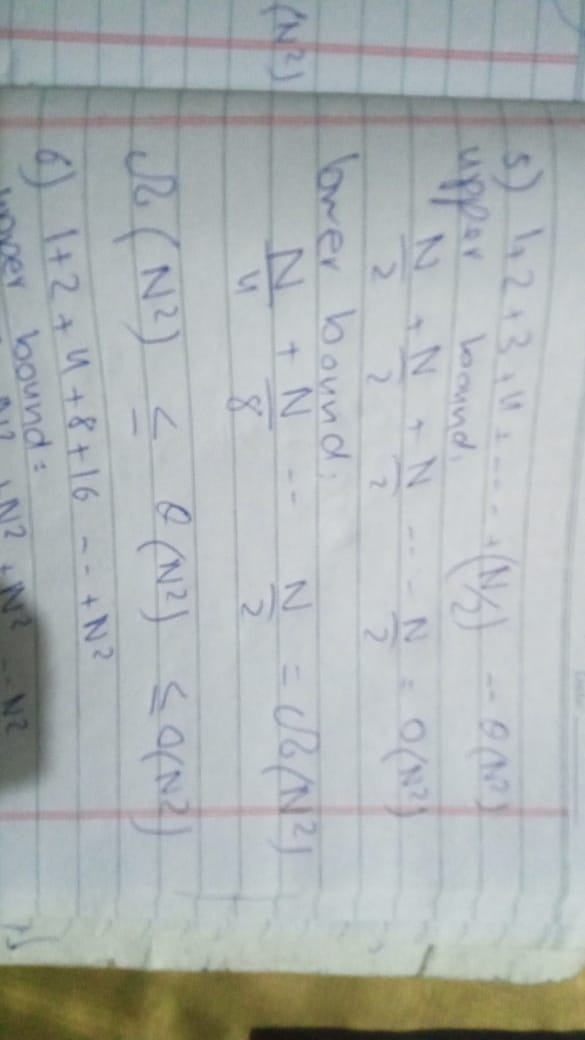
3)



4)



5)



6)

