**Merge Sort Implementation**

**And It’s Time Complexity**

**Analysis**

****

**Submitted To: Submitted By:**

**Md Arquam Gaurav Yadav**

**CSE**

**11911038**

* **Source Code:**

#include<stdio.h>

#include <stdlib.h>

void mergesort(int a[],int i,int j);

void merge(int a[],int i1,int j1,int i2,int j2);

int main()

{

int i;

int n=100;

int a[100];

for (int g = 0; g< n;g++)

{a[g] = (rand()%901)+100;}

mergesort(a,0,n-1);

printf("\nSorted array is :");

for(i=0;i<n;i++)

printf("%d ",a[i]);

return 0;

}

void mergesort(int a[],int i,int j)

{

int mid;

if(i<j)

{

mid=(i+j)/2;

mergesort(a,i,mid); //left recursion

mergesort(a,mid+1,j); //right recursion

merge(a,i,mid,mid+1,j); //merging of two sorted sub-arrays

}

}

void merge(int a[],int i1,int j1,int i2,int j2)

{

int temp[1000]; //array used for merging

int i,j,k;

i=i1; //beginning of the first list

j=i2; //beginning of the second list

k=0;

while(i<=j1 && j<=j2) //while elements in both lists

{

if(a[i]<a[j])

temp[k++]=a[i++];

else

temp[k++]=a[j++];

}

while(i<=j1) //copy remaining elements of the first list

temp[k++]=a[i++];

while(j<=j2) //copy remaining elements of the second list

temp[k++]=a[j++];

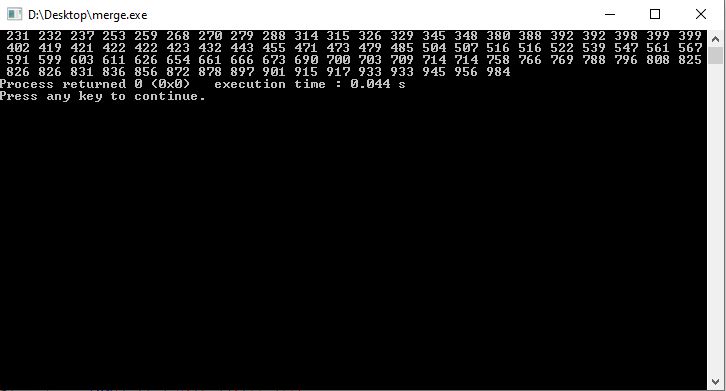
//Transfer elements from temp[] back to a[]

for(i=i1,j=0;i<=j2;i++,j++)

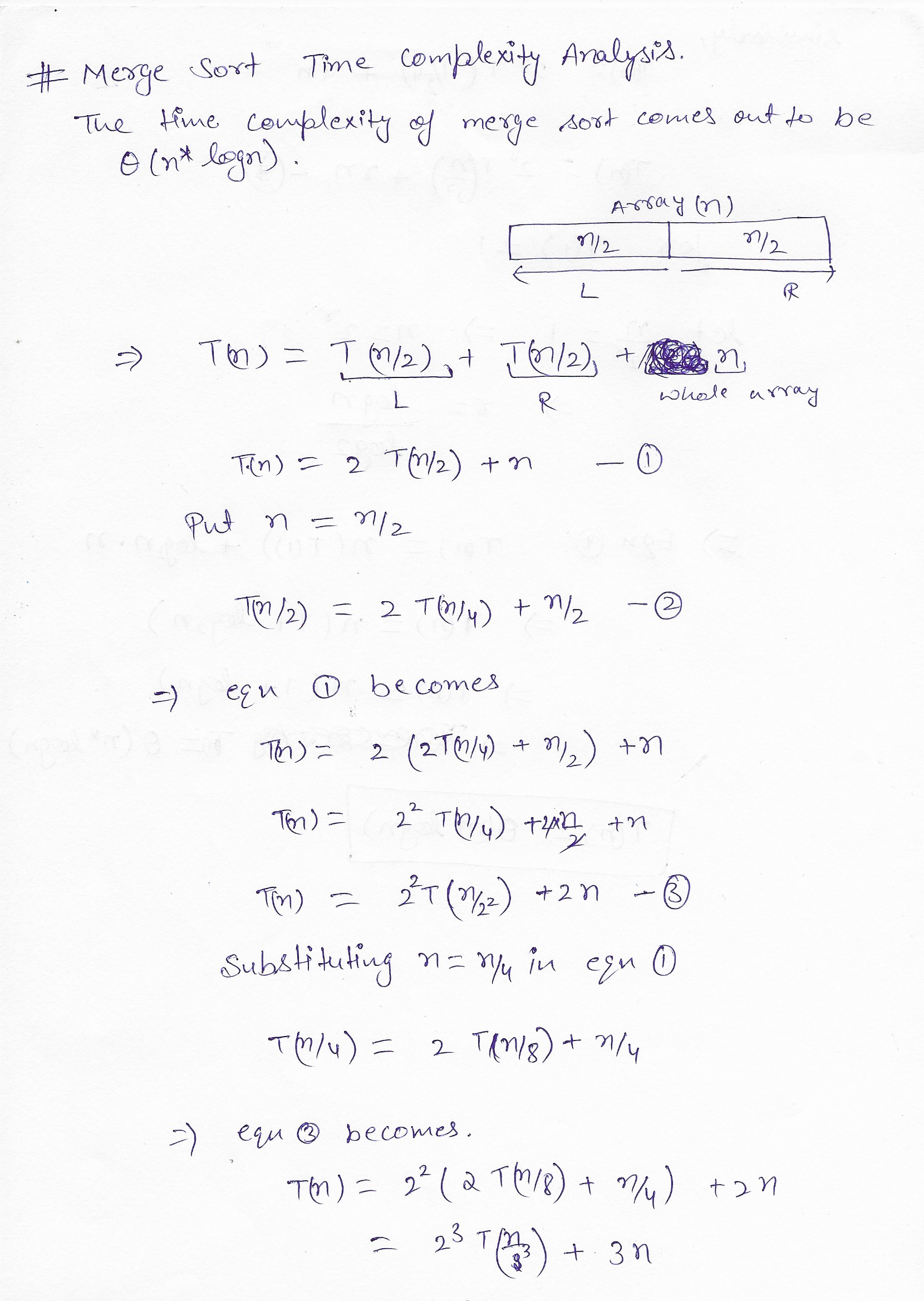
a[i]=temp[j];

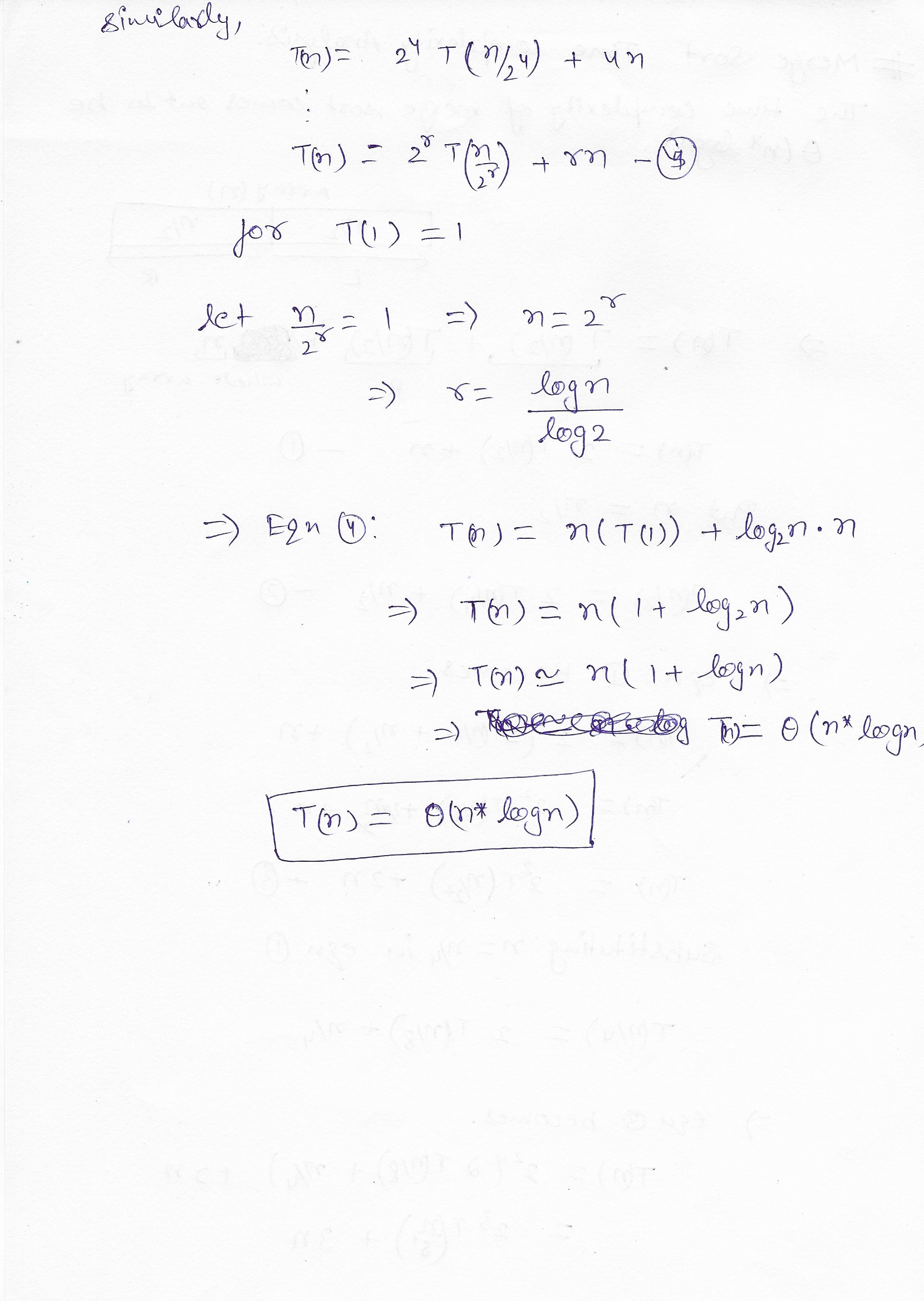
}

* **Output:**

****

* **Time Complexity Analysis:**

****

****