Design and Analysis of Algorithms

LAB EXPERIMENT-04

NAME:RIYA KUMARI

BATCH:34
SAPID:590016221

SUBMITTED TO:ARYAN SIR

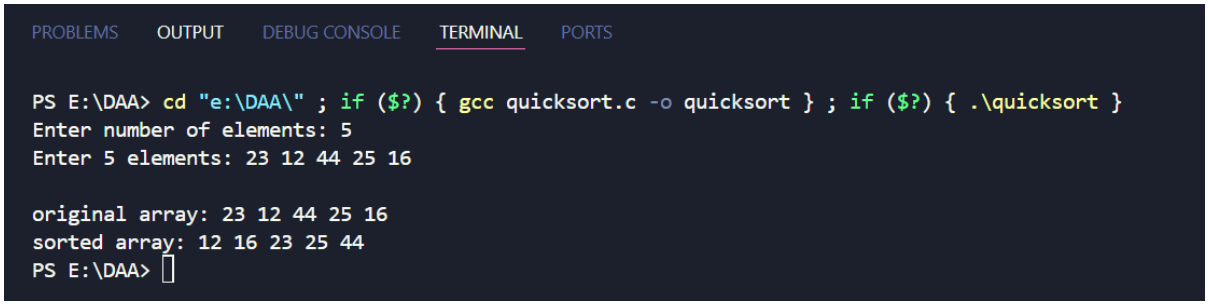
GITHUB REPOSITORY: https://github.com/Riyakumari1314/DAA-2nd-year

Implement Quick sort and analyze its time complexity.

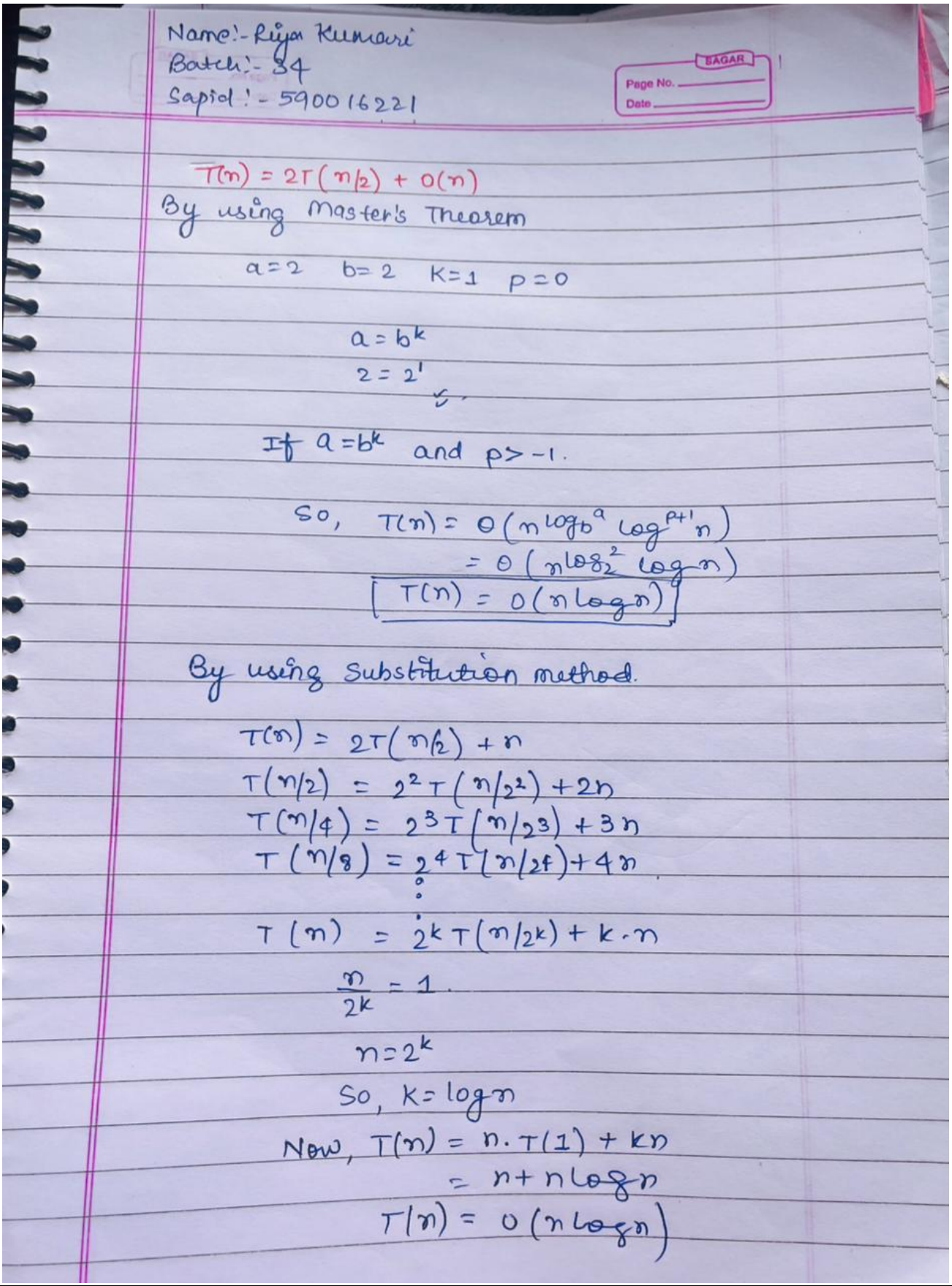
```
#include<stdio.h>

void swap(int*a,int*b){
	int temp=*a;
	*a=*b;
	*b=temp;
}
int partition(int arr[],int low,int high){
	int pivot=arr[high];
	int i=(low-1);
	for(int j=low;j<high;j++){
		if(arr[j]<pivot){
			i++;
			swap(&arr[i],&arr[j]);
		}
	}
	swap(&arr[i+1],&arr[high]);
	return (i+1);
}
void quicksort(int arr[],int low,int high){
	if(low<high){
		int pi=partition(arr,low,high);
		quicksort(arr,low,pi-1);
		quicksort(arr,pi+1,high);
	}
}
void printArray(int arr[],int size){
	for(int i=0;i<size;i++){
		printf("%d ",arr[i]);
	}
	printf("\n");
}
int main(){
	int n;
	printf("Enter number of elements: ");
	scanf("%d",&n);
	int arr[n];
	printf("Enter %d elements: ",n);
	for(int i=0;i<n;i++){
		scanf("%d",&arr[i]);
	}
	printf("\noriginal array: ");
	printArray(arr,n);
	quicksort(arr,0,n-1);
	printf("sorted array: ");
	printArray(arr,n);
	return 0;
}
```

Output-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS E:\DAA> cd "e:\DAA\" ; if ($?) { gcc quicksort.c -o quicksort } ; if ($?) { .\quicksort }
Enter number of elements: 5
Enter 5 elements: 23 12 44 25 16
original array: 23 12 44 25 16
sorted array: 12 16 23 25 44
PS E:\DAA>

Time complexity Analysis-



Name:- Riya Kumari
Batch:- 34
Sapid:- 590016221

$T(n) = 2T(n/2) + O(n)$

By using Master's Theorem

$a=2 \quad b=2 \quad K=1 \quad p=0$

$a = b^k$

$2 = 2^1$

If $a=b^k$ and $p > -1$.

So, $T(n) = O(n^{\log_b a} \log^{p+1} n)$

$= O(n^{\log_2^2} \log n)$

$[T(n) = O(n \log n)]$

By using Substitution method.

$T(n) = 2T(n/2) + n$

$T(n/2) = 2^2 T(n/2^2) + 2n$

$T(n/4) = 2^3 T(n/2^3) + 3n$

$T(n/8) = 2^4 T(n/2^4) + 4n$

$\vdots$

$T(n) = 2^k T(n/2^k) + k \cdot n$

$\frac{n}{2^k} = 1$

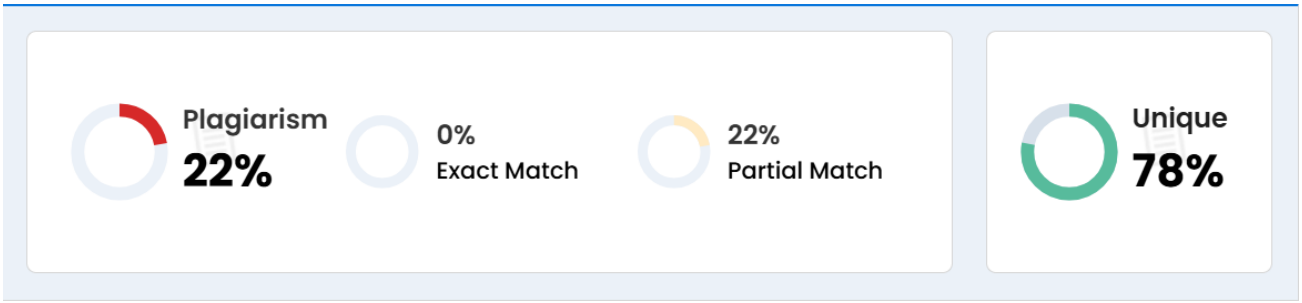
$n = 2^k$

So, $K = \log n$

Now, $T(n) = n \cdot T(1) + kn$

$= n + n \log n$

$T(n) = O(n \log n)$


Plagiarism
22%
0%
Exact Match
22%
Partial Match
Unique
78%