Q1 (15 Marks)

Solve the following recurrence, with all the steps and details and find the complexity of the recurrence. Show all detail.

$$T(N) = 5 * T\left(\frac{N}{5}\right) + N^{1/2} \qquad T(1) = 1$$

$T(N) = 5 * T(\frac{N}{5}) + N^{1/2}$

$= 5[5 * T(\frac{N}{25}) + (\frac{N}{5})^{1/2}] + N^{1/2}$

$= 25T(N/25) + 5\frac{N^{1/2}}{5^{1/2}} + N^{1/2}$

$= 25 * [5 * T(\frac{N}{125}) + (\frac{N}{25})^{1/2}] + 5^{1/2}N^{1/2} + N^{1/2}$

$= 125T(\frac{N}{125}) + 25(\frac{N^{1/2}}{25^{1/2}}) + 5^{1/2}N^{1/2} + N^{1/2}$

$= 5^3T(\frac{N}{5^3}) + \frac{25}{5}N^{1/2} + 5^{1/2}N^{1/2} + N^{1/2}$

$= 5^3T(\frac{N}{5^3}) + 5N^{1/2} + 5^{1/2}N^{1/2} + N^{1/2}$

$= 5^kT(\frac{N}{5^k}) + N^{1/2}[5^1 + 5^{1/2} + 5^0]$

$= 5^kT(\frac{N}{5^k}) + N^{1/2}\sum_{i=0}^{2}(5^{1/2})^i$

$= 5^{\log_5 N}T(1) + N^{1/2}\left[\sum_{i=0}^{k-1}(5^{1/2})^i\right]$

$= N^{\log_5 5} * 1 + N^{1/2}\left[\sum_{i=0}^{\log_5 N-1}(5^{1/2})^i\right]$

$= N * 1 + N^{1/2}[N^{1/2} - 1]$

$= N * 1 + N^1 - N^{1/2}$

$= O(N)$

$\frac{N}{5^k} = 1 \Rightarrow N = 5^k$

$\log N = \log 5^k$

$\log N = k \log 5$

$\boxed{\log_5 N = k}$

$\therefore \frac{(5^{1/2})^{\log_5 N / 1} - 1}{5^{1/2} - 1}$

$\therefore \frac{N^{\log_5 (5)^{1/2}} - 1}{5^{1/2} - 1}$ $\quad \therefore 5^{1/2} - 1 = C_1$

$\therefore \frac{N^{1/2} - 1}{C_1}$

Mid Solution

Q2 (25 Marks)

Given a sorted array, **Arr**, of size N, which contains only two distinct elements **x** & **y**, where **x < y**. It is required to find the count of **y** in the array **Arr**. **The Time Complexity of the algorithm shall be O($\log_2 n$). Keep in mind, function, naming convention, complexity, correct implementation and desired solution.**

a) Briefly describe your algorithm in words. (5 Marks)
b) Write a function that returns the count of **y** elements. (20 Marks)

Example:

Arr:	2 2 2 2 2 2 2 5 5 5 5	Returned Value:	4
Arr:	-12 -12 -12 7 7 7 7 7 7 7 7 7	Returned Value:	9

conceptual solution:

Function run with 2 arguments, Arr and N

We have two pointers one at start = 0 and one at the end = N-1

we will assigned x to the value

we will loop until start < end

  we will calculate mid

    if mid index value is equal to value move start to mid + 1

    if mid index value is greater than value and mid - 1 index value is value **then return the count**

    else end to mid -1

or **return count at the end**

```
int number_count_v1(int Arr[], int N){
    int count= 0, start = 0, end = N-1, mid = -1;
    int value = Arr[0];

    while(start < end){

    cout<<"At start "<<start<<" "<<end;
        mid = start + (end - start)/2;
        if(Arr[mid] == value){
            start = mid +1;
        }else if(Arr[mid] != value && Arr[mid-1] == value){
            return N - mid;
        }else{
            end = mid -1;
        }
        cout<<" at end "<<start<<" "<<end<<endl;
    }
    return N - end;
}
```

conceptual solution:

Function run with 2 arguments, Arr and N

We have two pointers one at start = 0 and one at the end = N-1

we will assigned x to the value

we will loop until start < end

  we will calculate mid

    if mid index value is equal to value move start to mid + 1

    else end to mid

or **return count at the end**

```
int number_count_v2(int Arr[], int N){
    int count= 0, start = 0, end = N-1, mid = -1;
    int value = Arr[0];

    while(start < end){

    cout<<"At start "<<start<<" "<<end;
        mid = start + (end - start)/2;
        if(Arr[mid] == value){
            start = mid + 1;
        }else{
            end = mid;
        }
        cout<<" at end "<<start<<" "<<end<<endl;
    }
    return N - end;
    return 0;
}
```