

1. Aşağıdaki kavramları kısaca açıklayınız:

- Big O,
- Big Theta,
- Big Omega,
- Best Case (En iyi Durum),
- Worst Case (En kötü durum),
- Excepted Case (Istisnai durum),
- Time Complexity (Zaman Karmaşıklığı),
- Space Complextiy (Alan Karmaşıklığı).

2. Aşağıdaki kod parçacığı ne işe yarar ve big O zamanı nedir?

```
int topN (int n)
{
    int sum = 0;
    while (n > 0) {
        sum += n % 10;
        n /= 10;
    }
    return sum;
}
```

3. Aşağıdaki kod parçacığı ne işe yarar ve çalışma süresi (runtime), zaman karmaşıklığı nedir?

```
int mod (int a, int b){
    if(b <= 0) {
        return -1;
    }
    int bol = a / b;
    return a - bol*b;
}
```

4. Aşağıda verilen kod için zaman karmaşıklığını $T(n)$, çalışma süresini hesaplayınız ve gerekçenizi açıklayınız.

```
void bubble_sort(int arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
    }
}
```

5. Aşağıda verilen kod için zaman karmaşıklığını $T(n)$ ve çalışma zamanını hesaplayınız ve gerekçenizi açıklayınız.

```
void merge(int arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;
    int L[n1], R[n2];

    for (i = 0; i < n1; i++) {
        L[i] = arr[l + i];
    }
    for (j = 0; j < n2; j++) {
        R[j] = arr[m + 1 + j];
    }
    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void merge_sort(int arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        merge_sort(arr, l, m);
        merge_sort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}
```