problem 24:

```
public void function_(int n) {
    int c = count = 0;
    for (i=1 ; i*i <=n , c++)
        count ++ ;
```

if $i^2 > n \Rightarrow T(n) = O(\sqrt{n})$.

problem 25.

```
public void function (int n) {
    int i, j, k, count = 0
    for (i = n/2; i <= n; i++) // thực hiện n/2 lần
        // thực hiện n/2 lần  for (j = 1; j + n/2 <= n; j++)
            // thực hiện log(n) lần  for (k=1, k <= n, k = k*2)
                count ++;
```

=> Độ phức tạp $O(n^2 \log n)$.

problem 26:

```
public void function (int n) {
    int i, j, k, count = 0
// Thực hiện n/2 lần  for (i = n/2, i <= n, i++)
// Thực hiện log2n lần  for (j=1, j<n, j = 2*j)
// thực hiện log n lần  for (k = 1, k <= n, k = k*2)
                count ++;
```

=> Độ phức tạp thuật toán: $O(n \log^2 n)$.

problem 27.

```
public void function(int n)
{ if (n == 1) return;
  for (int i = 1; i <= n; i++){
      for(int j = 1; j <= n; j++) {
          System.out.println("*");
```

Độ phức tạp thuật toán là $O(n)$