Asymptotic analysis Part II

```
function foo (n) {
    let ans = 0;                    ——— 1
    for ( let i = 1; i <= 2n; i++) {  — (n+1)
        for ( let j = 1; j <= n; j *= 2) {
            ans += 1;               ↳ n/2^k
        }                        log₂^n
    }
    console. log (ans);   — ①
}
```

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

$\Rightarrow n = 2^k$

$* \quad k = \log_2^n$

$T(n) = 1 + (n+1) + n * \log_2^n + L$

$T(n) = n \log_2^n + n + 3$

$T(n) = O(n \log_2^n)$

```
for ( let j = 1; j <= n; j *= 2) {
```