1. 计算

- 第一个循环遍历 1到2^6=64,每次循环i+=1
- 第二个循环遍历 1到i, 每次循环i*=2;
  - 每执行完一遍第二个for循环, res 增大了 $1 + [\log_2 i]$

因此程序执行完后, 有

$$res = 0 + \sum_{i \in [64]} 1 + [\log_2 i] = 7 + \sum_{j \in [6]} j \cdot 2^j = 328$$

2.时间复杂度

- (1) $T(n) = \Theta(n^{3.6})$
- (2) $T(n) = \Theta(3^n)$
- (3) $T(n) = \Theta(1)$
- (4) $T(n) = \Theta(3^n)$

$$T(n) + 1 = 3(T(n-1) + 1) = 3^{n-1}(T(1) + 1) = 2 \cdot 3^{n-1} \Rightarrow T(n) = \Theta(3^n)$$

3.

Proof:

(1)

- $b^n < a^n \Rightarrow b^n = O(a^n)$
- $\forall c$, 令 $n = \lceil \log_{a/b} c \rceil + 1$, 则有$a^n = (a/b)^n b^n > c \cdot b^n$, 因此 $a^n \neq O(b^n)$

(2)

- 上界
  由于

$$\sum_{k=1}^{n} \frac{1}{k} \leq 1 + \int_1^n \frac{1}{x} dx = 1 + \log n$$

  因此

$$\sum_{k=1}^{n} \frac{1}{k} = O(\log n)$$

- 下界
  由于

$$\int_1^{n+1} \frac{1}{x} dx = \log(n+1)$$

  由于

$$\frac{1}{k}$$

  是递减的，我们有：

$$\sum_{k=1}^{n} \frac{1}{k} \geq \int_1^{n} \frac{1}{x} dx = \log n$$

  因此：

$$\sum_{k=1}^{n} \frac{1}{k} = \Omega(\log n)$$

  联立上下界可知

$$\sum_{k=1}^{n} \frac{1}{k} = \Theta(\log n)$$

(3)
记 $f(x) = [\frac{n}{2}]$, 则有,

$$T(n) = T(f^{\lceil \log_2 n \rceil}(n)) + \lceil \log_2 n \rceil$$

注意到 $f(x) = [\frac{n}{2}] \leq n/2$, 因此有 $f^{\lceil \log_2 n \rceil}(n) \leq n(1/2)^{\lceil \log_2 n \rceil} \leq n(1/2)^{\log_2 n} = 1$, 由于 $f^{\lceil \log_2 n \rceil}(n) \in \mathbb{Z}_+$, 因此 $f^{\lceil \log_2 n \rceil}(n) = 0, 1$
规定 $T(0) = 0$, 不影响 $T(n)$ 条件的成立.从而,

$$T(n) \leq T(1) + \lceil \log_2 n \rceil = o(\log n)$$