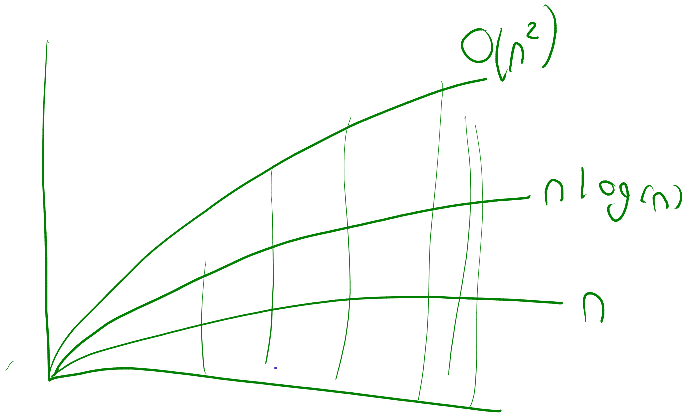
$$n < n\log(n) < n^2 < n^2\log(n) < n^3$$



$$\lim_{n\to\infty} n\log(n) \le 3n^2$$

$$\lim_{n\to\infty} \frac{n\log(n)}{n^2} \le \frac{3n^2}{n^2}$$

$$\boxed{0 \le 3}$$

$T(n) = T(n/2)+4$

$$T(n) = 4 + T(n/2)$$

$$T(n) = 4 + 4 + T(n/2^2)$$

$$4 \times 2 + T(n/2^2)$$

$T(n) = 8n + T(n/3)$

$$8n + \frac{8n}{3} + T(n/3^2) \quad T(n/3^3) \rightarrow T(n/3^4) = T\left(\frac{n}{3^k}\right)$$

$$1 = \frac{n}{3^k} \qquad 3^k = n \qquad \boxed{\log_3(n) = k}$$

$T(n) = 3n + 3T(n/3)$

$$T(n) = 3n + 3\left(\frac{\cancel{3}n}{\cancel{3}} + 3T(n/3^2)\right)$$

$$T(n) = 3n(k) + 3^k T(n/3^k)$$

$$1+k$$

$$T(n) = 3n + 3n + 3^2 T(n/3^2)$$

$$T(n) = 3n + 3n + 3^2\left(\frac{3n}{3^2} + 3T(n/3^3)\right)$$

$$T(n) = 3n + 3n + 3n + 3^3 T(n/3^3)$$

```
def algoritmo1(n):
    i = 2*n+1
    res = 0
    while i > 0:
      j = i*i
      res +=j
      i-=1
    return res
```

1

1

$2n+2$

$2n+1$

$2n+1$

$2n+1$

$S(i, res)$

$\boxed{S_0(2n+1, 0)}$

$(2n+1, 0) \to (2n, (2n+1)^2)$

$(2n-1, (2n+1)^2+(2n)^2)$

$(2n-2, (2n+1)^2+2n^2+(2n-1)^2+$

$(2n-3, (2n+1)^2+2n^2+(2n-1)^2+(2n-2)^2)$

$(0, \underbrace{(2n+1)^2+2n^2+\dots+1^2}) \to$

$(0, \sum_{k=1}^{2n+1} k^2)$

Invariante

$$\boxed{i, res = \sum_{k=i+1}^{2n+1} k^2}$$

$$\sum_{k=2n-2}^{2n+1} k^2 = (2n-2)^2+(2n-1)^2+(2n)^2+(2n+1)^2$$

Estado inical $\quad i=2n+1 \quad \sum_{k=2n+1+1}^{2n+1} k^2 = \sum_{k=2n+2}^{2n+1} k^2 = 0$

Estado Final $\quad i=0 \quad \sum_{k=0+1}^{2n+1} k^2 = \sum_{k=1}^{2n+1} k^2$

$(i, res) = (i-1, res+i^2)$

$\sum_{k=i+1}^{2n+1} k^2$

$\left(\sum_{k=i+1}^{2n+1} k^2\right)+i^2 \to i^2+(i+1)^2+(i+2)^2+\dots$

$\sum_{k=i}^{2n+1} k^2$

Reemplazar $\quad i=i-1 \quad \sum_{k=(i-1)+1}^{2n+1} k^2$

```
def algoritmo2(n):
    i = 0
    j = 0
    res = 0
    while i < 3*n:
        j = 2*i
        res -= j
        i+=1
    return res
```

1, 1, 1, $3n+1$, $3n$, $3n$, $3n$

$S\ (i, res)$ Estado

$S_0\ (0, 0)$ Estado inicial

$(0,0) \to (1,0) \to (2, 0-2) \to (3, 0-2-4) \to (4, 0-2-4-6)$

$-6 \qquad -12$

$\boxed{(i, res) \to (i+1, res-2*i)}$

Transformación

$(4, 0-2-4-6) \to (5, 0-2-4-6-8)$

$-12 \qquad -20$

$(6, -30)$

estado final

$i = 3n$

$(3n, 0-2-4-6-8-\dots-2(3n-1))$

$\boxed{\sum_{k=0}^{3n-1} 2k}$

$(2, 0-2)$

$-\sum_{k=0}^{1} 2k = -(2(0)+2(1)) = -2$

$(6, -30)$

$-\sum_{k=0}^{5} 2k = -(2(0)+2(1)+2(2)+(2)(3)+2(4)+(2)(5))$

$-(0+2+4+6+8+10) = -30$

10

Invariante

$res = -\sum_{k=0}^{i-1} 2k$

1) Estado inicial $(0,0)$ $-\sum_{k=0}^{-1} 2k = 0$

2) Estado final $\left(3n, \sum_{k=0}^{3n-1} 2k\right)$ reemplazamos invariante de ciclo $\sum_{k=0}^{3n-1} 2k$

3) Transformación estados

$(i, res) \to (i+1, res-2i)$

Reemplazando $i \to i+1$ $\quad -\sum_{k=0}^{i-1} 2k \to -\sum_{k=0}^{i} 2k$

Transformación $-\sum_{k=0}^{i-1} 2k - 2i$

$\underbrace{-(2(0)+2(1)+\dots+2(i-1))+2i}$

$-\left(\sum_{k=0}^{i} 2k\right)$

```
def algoritmo3(n):
    i = 4*n + 2
    j = 0
    res = 0
    while i > 0:
        while j<=3*n+4:
            res += 4
            j+=1
        i-=1
    return res
```

1
1
1
$4n+3$
$3n+6 + (1)(4n+2)$
$3n+5$
$3n+5$
$4n+2$

No entra

Entra

$j = 3n+5$

$(j, res) \to (j+1, res+4)$

(j, res) Forma estado

$(\boxed{0}, 0) \to (\boxed{1}, \boxed{0+4}) \to (\boxed{2}, \boxed{0+4+4}) \to (\boxed{3}, 0+4+4+4)$

$(0, 0)$ Estado inicial

$\boxed{(j, 4j) \text{ Invariante}}$

Estado inicial $(0,0) = (0, 4(0))$

Estado final $j = 3n+5$ $(3n+5, 4(3n+5))$

Transformacion $(j, res) \to (j+1, res+4)$

Invariante

$\boxed{res = 4j}$

$j = j+1$ $res = 4(j+1) = \boxed{4j+4}$

$res = res+4$ $\boxed{4j+4}$