



UNIVERSIDAD DIEGO PORTALES

Guía 1: Análisis y Algoritmos Aritméticos

YERKO ORTIZ
2025

1 Cotas superiores

Para cada una de las siguientes funciones, determine una cota superior utilizando la notación Big Oh $\mathcal{O}f(n)$.

1. $f(n) = n^2 + 50n$
2. $f(n) = 2^n + 3^n + 24n^{23}$
3. $f(n) = n^2 + 24n \log n$
4. $f(n) = 27 \log n + \sqrt{n}$
5. $f(n, m) = 5n + 8m$

2 Compilador a papel

Ejecute a mano cada uno de los siguientes algoritmos con las entradas especificadas y determine el output correspondiente.

1. Entero a Binario

```
static String intToBin(int N) {  
    String bin = "";  
    bin += N%2;  
    while(N > 1) {  
        N/=2;  
        bin = N%2 + bin;  
    }  
    return bin;  
}
```

- a. $N = 43$
- b. $N = 176$
- c. $N = 1092$

2. Máximo común divisor

```
static int gcd(int a, int b) {  
    int res = a%b;  
    while(res != 0) {  
        a = b;  
        b = res;  
        res = a%b;  
    }  
    return b;  
}
```

- a. $a = 3213, b = 2$
- b. $a = 2046, b = 360$
- c. $a = 2046, b = 1024$
- d. $a = 97, b = 47$

3. Criba de Eratóstenes

```
static boolean[] eratosthenes(int n) {  
    boolean[] primes = new boolean[n];  
    for(int i = 0; i < n; i++){  
        primes[i] = true;  
    }  
  
    for (int p = 2; p * p <= n; p++) {  
        if (primes[p] == true) {  
            for (int i = p * p; i <= n; i += p)  
                primes[i] = false;  
        }  
    }  
  
    return primes;  
}
```

- a. $N = 32$

3 Análisis

Para cada uno de los siguientes métodos determine el tiempo de complejidad utilizando la notación Big Oh $\mathcal{O}f(n)$.

1.

```
static void f1(int N){  
    for(int i = 0; i < N; i++) {  
        // <body>  
    }  
}
```

2.

```
static void f2(int N){  
    for(int i = 0; i < N; i++) {  
        for(int j = 0; j < N; j++) {  
            // <body>  
        }  
    }  
}
```

3.

```
static void f3(int N){  
    for(int i = 1; i < N; i*=2) {  
        // <body>  
    }  
}
```

4.

```
static void f4(int N){  
    for(int i = 0; i < N; i++) {  
        for(int j = 1; j < N; j*=2) {  
            // <body>  
        }  
    }  
}
```

5.

```
static void f5(){  
    for(int i = 0; i < 1024; i++) {  
        // <body>  
    }  
}
```

6.

```
static void f6(int N){
    for(int i = 0; i < N; i+=2) {
        // <body>
    }
}
```

7.

```
static void f7(int N){
    for(int i = 0; i < N*N; ++i) {
        // <body>
    }
}
```

8.

```
static void f8(int N){
    for(int i = 0; i*i < N; ++i) {
        // <body>
    }
}
```

9.

```
static void f9(int N){
    for(int i = 0; i*i < N*N; ++i) {
        // <body>
    }
}
```

10.

```
static void f10(int N, int M){
    for(int i = 0; i < N; ++i) {
        // <body>
    }

    for(int i = 0; i < M; ++i) {
        // <body>
    }
}
```

11.

```
static void f11(int N, int M){
    for(int i = 0; i < N; ++i) {
        for(int j = 0; j < M; ++j) {
            // <body>
        }
    }
}
```

12.

```
static void f12(int N){
    for(int i = 0; i < N; ++i) {
        for(int j = 0; j < 100; ++j) {
            // <body>
        }
    }
}
```

13.

```
static void f13(int N){
    while(N > 0) {
        // <body>
        --N;
    }
}
```

14.

```
static void f14(int N){
    while(N > 0) {
        // <body>
        N/=2;
    }
}
```

15.

```
static void f15(int N){
    while(N > 0) {
        // <body>
        N/=10;
    }
}
```

4 Problemas

4.1 Número palíndromo

Diseñe un algoritmo que reciba como entrada un número entero N y retorne como salida un booleano que describe si los dígitos (base 10) que conforman a N forman un palíndromo.

Entrada

- i. Número entero N .

Restricciones

- i. $1 \leq N \leq 10^9$

Salida

- i. Booleano que describe si los dígitos que componen a N en sistema decimal conforman un palíndromo.

Ejemplos

- a. Entrada: 1001
Salida: true
- b. Entrada: 234
Salida: false

4.2 Factores Primos

Diseñe un algoritmo que reciba como entrada un número entero N y como salida retorne los factores primos que componen a N , es decir el producto de números primos cuyo resultado es N .

Entrada

- i. Número entero N .

Restricciones

- i. $1 \leq N \leq 10^6$

Salida

- i. Arreglo de enteros que contiene los factores primos de N .

Ejemplos

- a. Entrada: 105
Salida: $\{3, 5, 7\}$
- b. Entrada: 32
Salida: $\{2, 2, 2, 2, 2\}$

4.3 Decimal a hexadecimal

Diseñe un algoritmo que reciba como entrada un número entero N (base 10) y retorne como salida su representación de tipo String en sistema hexadecimal (base 16).

Hint: Dado que el sistema hexadecimal requiere dígitos de 0 hasta 15, utilice el alfabeto desde la A para representar dígitos que en sistema decimal requerirían dos dígitos, 10 se representa con la letra A, 11 se representa con la letra B, 12 con la letra C, 13 con la letra D, 14 con la letra E y 15 con la letra F.

Entrada

- i. Número entero N .

Restricciones

- i. $1 \leq N \leq 10^9$

Salida

- i. String que contiene la representación de N en hexadecimal.

Ejemplos

- a. Entrada: 15
Salida: $\{F\}$
- b. Entrada: 32
Salida: $\{2, 0\}$

4.4 Consultas de dígitos

Considere la existencia de un String infinito conformado por la concatenación de todos los números naturales (enteros positivos) en orden ascendente:

12345678910111213141516171819202122232425...

Diseñe un algoritmo que recibe como entrada un entero q que describe la cantidad de consultas que su algoritmo debe contestar, seguido de q líneas, donde cada línea contiene un número entero q_i , este describe la i -ésima posición en el String de dígitos.

Como output su algoritmo debe retornar una línea por consulta, donde en cada línea debe estar el i -ésimo dígito del String, es decir el dígito en la posición q_i . Considere que el String se indexa desde 1 hacia el infinito.

Entrada

- i. Número entero q .
- ii. Secuencia de q consultas q_i .

Restricciones

- i. $1 \leq q \leq 10^3$
- ii. $1 \leq q_i \leq 10^{18}$

Salida

- i. Una línea por consulta, donde en cada línea se describe el dígito en la posición q_i

Ejemplos

- a. Entrada: 1

1

Salida: 1

- b. Entrada: 6

1

3

5

10

11

19

Salida: 1

3

5

1

0

4

Hint: para resolver este problema use lo aprendido sobre divisibilidad y sistemas posicionales y así encontrar una expresión matemática que le permita encontrar el dígito.