

Adivinar un número natural^{*}

Juan David Hernandez Rodriguez¹

^a*Pontificia Universidad Javeriana, Bogotá, Colombia*

Abstract

En este documento se presenta la escritura formal del problema «adivinar un número natural»

1. Análisis del problema

Para la solución a este problema se utilizará la técnica llamada dividir y vencer, la cual es una técnica utilizada en resolver algoritmos; esta consiste en dividir el problema en problemas más pequeños, así para hallar una solución combinando todas las soluciones individuales de los subproblemas. Para realizarlo de esta manera, se requiere de 3 pasos.

- Dividir el problema en problemas más pequeños, usualmente se utiliza 1 o varios pivotes para saber dónde se va a dividir.
- Resolver cada uno de los subproblemas aparte, y a cada subproblema se aplica la misma estrategia de manera recursiva.
- Por último, debemos combinar las soluciones obtenidas de los problemas más pequeños para así obtener la solución final.

Para saber qué tan rápido es el algoritmo con este método se puede calcular la complejidad utilizando la recurrencia, la cual describe el número de operaciones en términos del tamaño del problema original. Se utiliza la siguiente fórmula para calcularla.

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)$$

donde a representa el número de subproblemas generados en cada división, b representa el tamaño de cada subproblema y $f(n)$ representa el costo de dividir el problema original, resolver subproblemas y combinar soluciones, o mucho más sencillo es todo lo que no sea recursión en el algoritmo.

^{*}Este documento presenta la escritura formal de un algoritmo.

Email address: juan_hernandezr@javeriana.edu.co (Juan David Hernandez Rodriguez)

2. Diseño del problema

El análisis anterior nos permite diseñar el problema: definir las entradas y salidas de un posible algoritmo de solución, que aún no está definido.

1. Entradas: Se necesitan 3 parametros de entrada, el limite inferior del rango, el timite superior del rango definido y los posibles valores a adivinar.
2. Salidas: El numero del usuario adivinado

3. Algoritmos de solución

3.1. Algoritmo recursivo

Algorithm 1 Algoritmo recursivo para adivinar un numero.

```

1: procedure ADIVINARDIVIDIRVENCER( $b, e, n$ )
2:    $numeros \leftarrow []$ 
3:   if  $e = b$  then
4:     return  $b$ 
5:   end if
6:   for  $i \leftarrow 1$  to  $n$  do
7:      $guess \leftarrow \text{randint}(b, e)$ 
8:      $numeros \leftarrow numeros + \text{randint}(b, e)$ 
9:     print “El número, guess, es Mayor(1), Menor(2) o Igual(0)?”
10:     $inp \leftarrow \text{input}()$ 
11:    if  $inp = 1$  then
12:       $e \leftarrow guess - 1$ 
13:    else if  $inp = 2$  then
14:       $e \leftarrow guess + 1$ 
15:    else if  $inp = 0$  then
16:      return  $guess$ 
17:    end if
18:  end for
19:  return  $AdivinarDividirVencer(b, e, n)$ 
20: end procedure
21: procedure ADIVINARNUMERO
22:   while  $True$  do
23:     print “Dame el rango superior”
24:      $e \leftarrow \text{input}()$ 
25:     print “Dame la cantidad de numeros a adivinar”
26:      $n \leftarrow \text{input}()$ 
27:     if  $e \geq 0$   $n \geq 0$  then
28:       print “Piensa un número del 0 al, e”
29:       print “print(AdivinarDividirVencer(0, e, n), es tu número!)”
30:       break
31:     else
32:       end if
33:   end while
34: end procedure

```

3.1.1. Análisis de complejidad

Reemplazando en la formula ya descrita anteriormente para conocer la complejidad de un algoritmo de dividir y vencer se tiene lo siguiente:

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

ahora aplicando el teorema maestro se logra ver que $\log_b a = \log_2 1 = 0$.
Por lo tanto la complejidad del algoritmo es:

$$T(n) = \Theta(\log n)$$