



## TALLER: DIVIDE Y VENCERAS

ANDERSON JAIR ALVARADO

RESUMEN. En este documento se presenta el problema para hallar la máxima  $i$ -diferencia de una secuencia. Esto se llevará a cabo con la escritura básica de los algoritmos: análisis, diseño y los algoritmos Iterativo y Divide y Vencerás que solucionen el problema.

### ÍNDICE

<b>Parte 1. Análisis y Diseño</b>	1
1. Análisis	1
2. Diseño	2
<b>Parte 2. Algoritmos y Complejidad</b>	2
3. Algoritmo Iterativo	2
4. Complejidad	3
5. Algoritmo Divide y Vencerás	3
6. Complejidad	4
<b>Parte 3. Análisis Experimental</b>	4
7. Prueba de Complejidad	4
8. Prueba de Funcionalidad	5

### Parte 1. Análisis y Diseño

#### 1. ANÁLISIS

Dada una secuencia  $S = \langle s_1, s_2, s_3, \dots, s_n \rangle$ , donde  $s_i \in \mathbb{Z}$ , se define la  $i$  Diferencia como:  $D_i = \text{sum}(s_1, \dots, s_{i-1}) - \text{sum}(s_{i+1}, \dots, s_n)$ .

## 2. DISEÑO

Entradas:

1. Una secuencia  $S$  de  $n$  elementos:  $S = \langle s_1, s_2, s_3, \dots, s_n \rangle$ , donde  $s_i \in \mathbb{R}$ .

Salidas:

1. Un valor  $v$  de la mayor  $i$  Diferencia:  $D_i = \text{sum}(s_1, \dots, s_{i-1}) - \text{sum}(s_{i+1}, \dots, s_n)$  dentro de la secuencia  $S$ .

## Parte 2. Algoritmos y Complejidad

## 3. ALGORITMO ITERATIVO

De forma itereativa se halla la mayor  $i$ -Diferencia dentro de la secuencia  $S$ .

---

**Algorithm 1** Máxima i-Diferencia Iterativo.

---

```

1: procedure MAX_I_DIFERENCIA( $S$ )
2:    $max \leftarrow -\infty$ 
3:   for  $i \leftarrow 1$  to  $|S|$  do
4:     if  $i = 1$  then
5:        $sum\_izquierda \leftarrow 0$ 
6:        $sum\_derecha \leftarrow \text{sum}(S, i + 1, |S|)$ 
7:     end if
8:     else
9:       if  $i = |S|$  then
10:         $sum\_izquierda \leftarrow \text{sum}(S, 1, i - 1)$ 
11:         $sum\_derecha \leftarrow 0$ 
12:       end if
13:     end else
14:     else
15:        $sum\_izquierda \leftarrow \text{sum}(S, 1, i - 1)$ 
16:        $sum\_derecha \leftarrow \text{sum}(S, i + 1, |S|)$ 
17:     end else
18:
19:      $dif \leftarrow sum\_izquierda - sum\_derecha$ 
20:     if  $dif > max$  then
21:        $max \leftarrow dif$ 
22:     end if
23:   end for
24:   return  $max$ 
25: end procedure
26:
27: procedure SUM( $S, b, e$ )
28:    $suma \leftarrow 0$ 
29:   for  $i \leftarrow b$  to  $e$  do
30:      $suma \leftarrow suma + S_i$ 
31:   end for
32:   return  $suma$ 
33: end procedure

```

---

## 4. COMPLEJIDAD

Siendo  $|S| = n$ , el algoritmo *Máxima i-Diferencia Iterativo* tiene orden de complejidad  $O(n^2)$  debido al ciclo for presente, y el ciclo for presente en la función **sum()** que se iteran en el peor de los casos un total de  $n^2$  veces.

## 5. ALGORITMO DIVIDE Y VENCERÁS

De forma Recursiva mediante Divide y Venceras se halla la mayor i-Diferencia.

---

**Algorithm 2** Máxima i-Diferencia Divide y Vencerás.

---

```

1: procedure MAX_I_DIFERENCIA( $S$ )
2:    $max \leftarrow -\infty$ 
3:   for  $i \leftarrow 1$  to  $|S|$  do
4:     if  $i = 1$  then
5:        $sum\_izquierda \leftarrow 0$ 
6:        $sum\_derecha \leftarrow \text{sum}(S, i + 1, |S|)$ 
7:     end if
8:   else
9:     if  $i = |S|$  then
10:       $sum\_izquierda \leftarrow \text{sum}(S, 1, i - 1)$ 
11:       $sum\_derecha \leftarrow 0$ 
12:    end if
13:  end else
14:  else
15:     $sum\_izquierda \leftarrow \text{sum}(S, 1, i - 1)$ 
16:     $sum\_derecha \leftarrow \text{sum}(S, i + 1, |S|)$ 
17:  end else
18:
19:   $dif \leftarrow sum\_izquierda - sum\_derecha$ 
20:  if  $dif > max$  then
21:     $max \leftarrow dif$ 
22:  end if
23: end for
24: return  $max$ 
25: end procedure
26:
27: procedure SUM( $S, b, e$ )
28:   if  $b > e$  then
29:     return 0
30:   end if
31: else
32:    $q \leftarrow \lfloor (b + e) \div 2 \rfloor$ 
33:   return  $\text{sum}(S, b, q-1) + S_q + \text{sum}(S, q+1, e)$ 
34: end else
35: end procedure

```

---

## 6. COMPLEJIDAD

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

$$1 \in O(n^{\log_2(2-\epsilon)}) \rightarrow T(n) \in \Theta(n^{\log_b a}) \rightarrow \log_2(2-\epsilon) = 0 : \epsilon = 1$$

$$\Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$$

Por lo tanto la complejidad de todo el algoritmo sería  $O(n^2)$  en el peor de los casos.

## Parte 3. Análisis Experimental

Para la parte experimental de taller se experimentó con varios datos para analizar la complejidad y el funcionamiento del algoritmo.

## 7. PRUEBA DE COMPLEJIDAD

Se realizó una prueba con 30000 secuencias y se realizó una gráfica contra el tiempo para verificar la complejidad algorítmica del algoritmo *Máxima i-Diferencia Divide y Vencerás*.

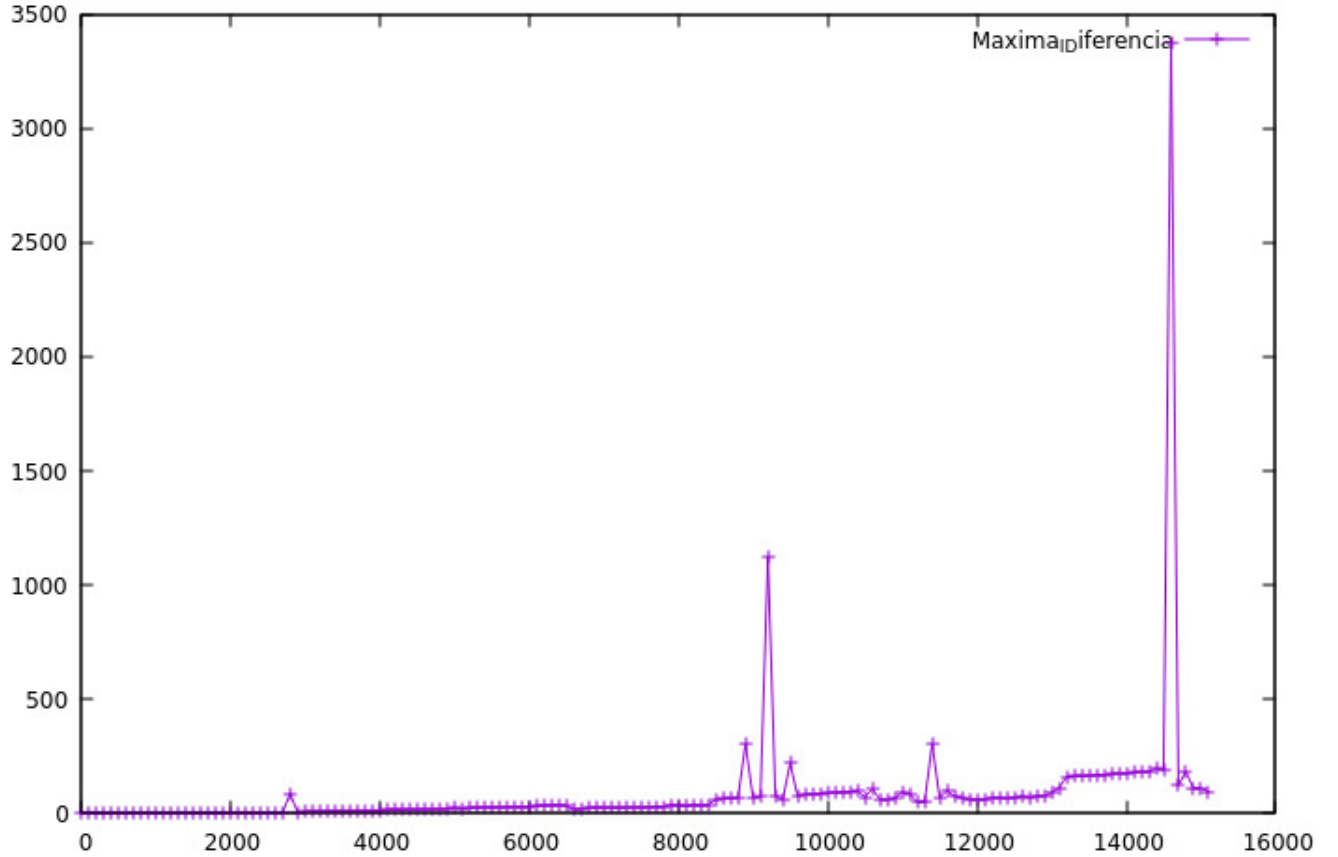


FIGURA 1. Gráfica resultado.

## 8. PRUEBA DE FUNCIONALIDAD

Con 10000 (diez mil) datos de entrada de secuencias de diferente tamaño del archivo `test_cases.in` y los datos esperados de salida del archivo `test_cases.out`, se puso a prueba la funcionalidad del algoritmo comparando que los datos de salida fueran iguales.

Se creó un archivo `result_cases.txt` con los siguientes datos de las columnas.

- **Valor Esperado:** Representa el valor de salida que debe dar al usar el algoritmo, que se encuentran en el archivo `test_cases.out`.
- **Valor Obtenido:** Representa el valor experimental, el valor obtenido al usar el algoritmo con las secuencias de entrada dadas.

Algunos de los primeros datos del archivo de comparación generado:

Valor Deseado	Valor Obtenido
281.29012977435616	281.29012977435616
156.48568992703397	156.48568992703397
70.24350355808531	70.24350355808531
26.108480283212373	26.108480283212373
614.9450625546652	614.9450625546652
205.02325442087377	205.0232544208738
223.5678353629532	223.5678353629532
178.28556243611672	178.2855624361167
242.5266758908977	242.52667589089774
550.0285285014244	550.0285285014243
98.1103443028035	98.11034430280353
-30.815347601947565	-30.815347601947565
629.5142871912747	629.5142871912747
229.31677392217273	229.31677392217273
609.5125265878057	609.5125265878057
336.70927665458044	336.70927665458044
413.5558921070754	413.5558921070755
271.1107636776329	271.1107636776329
563.9400969908486	563.9400969908485
335.0115013301042	335.0115013301042
207.69874015089653	207.69874015089658
252.08722025428614	252.08722025428608
175.4056179130265	175.40561791302648
350.02528040550544	350.02528040550544
692.1232347519004	692.1232347519004
334.1603835765963	334.1603835765964
106.44476449661175	106.44476449661177
318.2835117939243	318.28351179392433
11.03702048577469	11.037020485774704
335.9075355029514	335.90753550295136
377.27223480885164	377.2722348088514
1102.4085088068557	1102.4085088068557
70.00518331547813	70.00518331547812

FIGURA 2. Valor Esperado vs Valor Obtenido.

El valor obtenido es igual al valor esperado, variando solo en cifras decimales, por ende se concluye un correcto funcionamiento del algoritmo *Máxima i-Diferencia Divide y Vencerás*.