Pontifica Universidad Católica del Perú

Trabajo Académico del curso Arquitectura de Computadoras

Tiempo de ejecución y optimización

Alumno

Alejandro Macedo Pereira

Profesor

Jorge Benavides Aspiazu

24 de noviembre de 2017

1. Introducción

2. Primer algoritmo

El primer algoritmo usado para resolver el problema fue tratar de mantener la mínima distancia entre los 3 números, lo cual se consigue avanzando al término siguiente del menor de los 3.

Triangulares	Pentagonales	Hexagonales
3	5	6
6	5	6
10	**5**	6
10	12	**6**
	.	
	.	
	.	

Cuadro 1: Primer algoritmo seleccionando el menor de los 3

Como se sabe que estos 3 números en algún momento serán iguales, este algoritmo terminará cuando se cumpla esta condición. Una porción del código será presentado a continucación (*e*l código completo se encuentra en la carpeta PrimeraForma).

```
 8 def hallar():
 7     nNums = [286, 165, 143]
 6     nums = [hallarTri(nNums[0]), hallarPen(nNums[1]), hallarHex(nNums[2])]
 5
 4     finish = True
 3     while finish:
 2         if igu(nums[0], nums[1], nums[2]):
 1             break
31
 1         m = hallarMin(nums)
 2         nNums[m] = nNums[m] + 1
 3         if m == 0:
 4             nums[0] = hallarTri(nNums[m])
 5         elif m == 1:
 6             nums[1] = hallarPen(nNums[m])
 7         elif m == 2:
 8             nums[2] = hallarHex(nNums[m])
 9
```

Figura 1: Porción de código que ejecuta el algoritmo antes mostrado

Adicionalmete se ha desarrollado el mismo algoritmo pero en C++ con el motivo de servir de comparación entre estos dos lenguajes. Los resultados de ambos códigos se presentan a continuación:

Cuadro 2: Tiempos de ejecución para ambos lenguajes

	Python	C++
Tiempo	130ms - 140ms	3ms - 6ms

Como se observa, el tiempo que demora Python en ejecutar el código no es muy alto, lo que nos indica que el algoritmo usado es eficiente ya que este tiempo está en el orden de los milisegundos; sin embargo no se compara a C++ ya que toma aproximadamente $\frac{1}{10}$ del tiempo.

3. Segundo algoritmo

Luego de investigar en Internet, ví que este problema había sido planteado en la página de Project Euler, en la cual se presentan desafíos de matemática y programación [?], siendo este el problema número 45. De la misma forma también encontré una solución a este desafío hecha por Kristian Edlund en su página Mathblog [?] donde presenta una solución aprovechando la propiedad de que el conjunto de los números hexagonales es un subconjunto de los números triangulares, lo cual se prueba a continuación.

$$\begin{gathered} T(n) = n * (n+1)/2 \\ reemplazando\ n = 2m-1 \\ T(2m-1) = (2m-1) * ((2m-1)+1)/2 \\ T(2m-1) = (2m-1) * 2m/2 \\ T(2m-1) = m * (2m-1) = H(m) \end{gathered} \tag{1}$$

Esto facilita mucho el cálculo del número pedido ya que solo se necesitaría saber si un número hexagonal es al mismo tiempo pentagonal. Esto se consigue con la inversa de la fórmula de los números pentagonales; si n es un número entero, entonces también es pentagonal.

$$n = \frac{\sqrt{24x+1}+1}{6} \tag{2}$$