

Taller en Sala No. 7
Notación O con recursión

INTEGRANTES:

- Gonzalo García Hernández
- Sebastián Henao Zapata

Vida real: En la vida real, la serie de Fibonacci se utiliza para diseñar la distribución de objetos en los videojuegos de tal forma que las escenas cumplan con una cierta estética determinada por la proporción aurea de Fibonacci
https://i.ytimg.com/vi/QQoj2_4SvX4/maxresdefault.jpg

Para resolver en parejas.

Considere los siguientes algoritmos.

ArrayMax, calcula el máximo valor de un arreglo de enteros. A es el arreglo y n es la última posición del arreglo en el primer llamado. ArrayMax se llama recursivamente hasta que n sea igual a 0.

Pista: Como un ejemplo ArrayMax([1,2,3,8,10,4], 5) retorna 10.

Pista 2: El tamaño del problema aquí es el número de elementos del arreglo

```
SubProceso ArrayMax( A, n )
    Definir i, max, temp Como Entero;
    max <- A[n]; // Si n = 0, max <- A[0]
    Si n != 0 Entonces
        temp <- ArrayMax(A, n-1);
        Si temp > max Entonces
            max <- temp;
    Retornar max;
```

El algoritmo **groupSum**, dado un arreglo de enteros, decide si es posible escoger un subconjunto de esos enteros, de tal forma que la suma de los elementos de ese subconjunto sea igual al parámetro **target**. El parámetro **start** funciona como un contador y representa un índice en el arreglo de números **nums**.

Pista: El tamaño del problema aquí es el número de elementos del arreglo

```
public boolean groupSum(int start, int[] nums, int target) {
    if (start >= nums.length) return target == 0;
    return groupSum(start + 1, nums, target - nums[start])
        || groupSum(start + 1, nums, target);
}
```

```
}
```

El algoritmo fibonacci calcula el valor enésimo de la serie de Fibonacci recursivamente.

Pista: El tamaño del problema es el término enésimo (int n).

```
public static long fibonacci(int n) {
    if (n <= 1) return n;
    else return fibonacci(n-1) + fibonacci(n-2);
}
```

1. Implementar el algoritmo de ArrayMax en Java y copiar los otros dos.
2. Identificar qué representa el tamaño del problema para cada algoritmo

ArrayMax: a.length
 Groupsum: nums.length
 Fibonacci: int n

3. Identificar valores apropiados para tamaños del problema

ArrayMax: 5, 10, 15, 20,25,30,35,40,45,50
 GroupSum: 5, 10, 15, 20,25,30,35,40,45,50
 Fibonacci: 20,22,24,26,28,30,32,34,36,38

4. Tomar los tiempos para los anteriores algoritmos con 10 tamaños del problema diferentes.

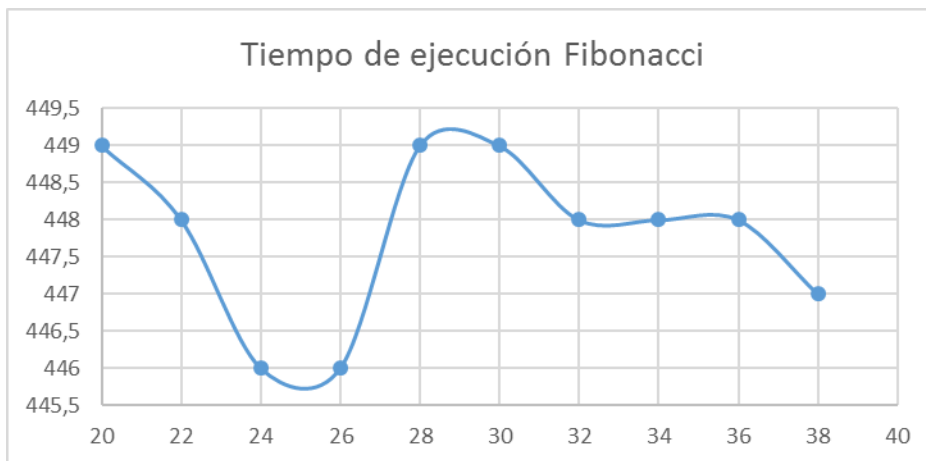
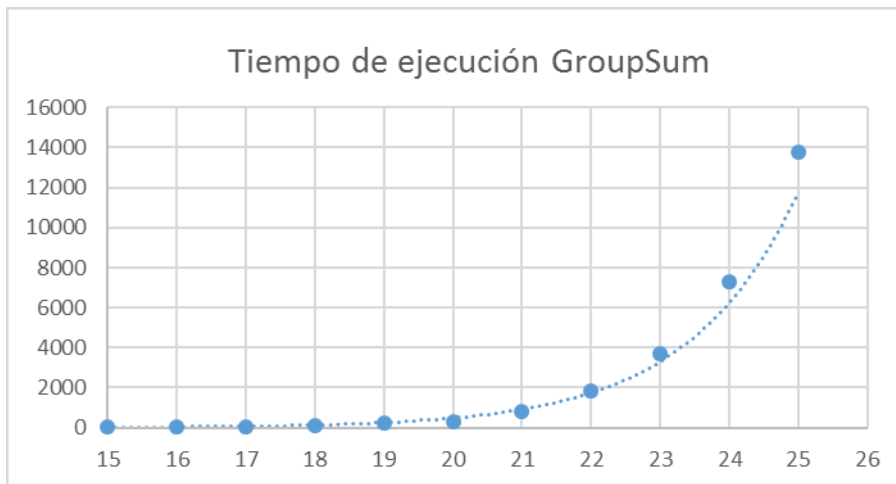
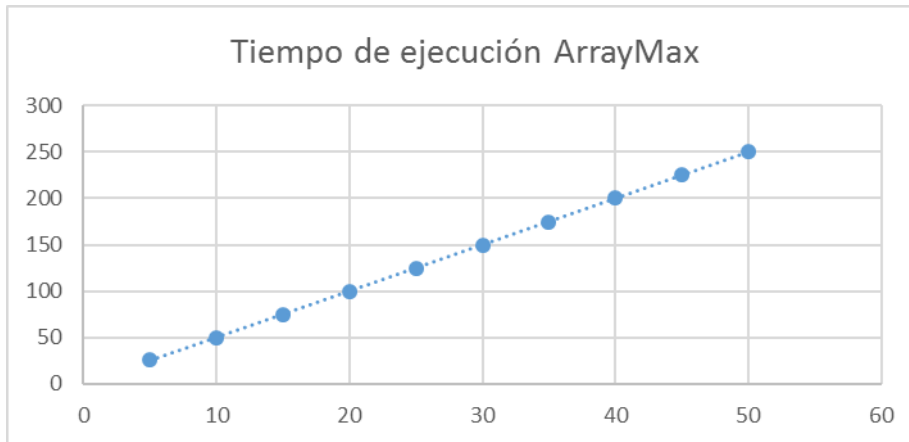
Pista: La sección 4.2 de la Guía de Laboratorios muestra cómo generar arreglos con números aleatorios

Pista 2: La sección 4.3 de la Guía de Laboratorios muestra cómo aumentar el heap en Java y la sección 4.4 muestra cómo aumentar el tamaño de la pila.

Pista 3: La sección 4.7 de la Guía de Laboratorios muestra cómo tomar el tiempo en Java

5. Hacer una gráfica en Excel para cada algoritmo y pasarla a Word luego

Pista 4: La sección 4.6 de la Guía de Laboratorios muestra cómo cambiar la escala de una gráfica en Excel a escala logarítmica.



6. Entregar el archivo de Word pasado a pdf. No entregar el código.