



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Análisis de algoritmos
Parcial 1 - Agosto 17 de 2023 (9am-11am)
Nombre:

Reglas y recomendaciones

Durante el parcial, deben observarse las siguientes reglas:

1. El parcial es estrictamente individual y se debe desarrollar en el salón de clase.
2. **Las únicas respuestas válidas serán las contenidas las cajas en blanco del presente parcial.**
3. Está absolutamente prohibido el uso de cualquier herramienta de comunicación (digital o análoga) entre usted y cualquier otro ser humano.
4. **Toda pregunta acerca del enunciado del parcial se debe hacer desde el puesto en voz alta.**

1. (20 %) Análisis de un algoritmo recurrente

Considere la siguiente estructura algorítmica:

```
1: procedure FuBaR_GtO( $M, b, e$ )  
Require:  $M = \langle m_i \in \mathbb{R} \rangle$   
Require:  $b \in \mathbb{N} \leq e \in \mathbb{N}$  son los límites de la secuencia  $M$ .  
2:   if  $b > e$  then  
3:     return  $\emptyset$   
4:   else  
5:      $q \leftarrow \lfloor (b + e) \rfloor \div 2$   
6:      $x \leftarrow \text{FuBaR\_GtO}(M, b, q - 1)$   
7:      $y \leftarrow \text{FuBaR\_GtO}(M, q + 1, e)$   
8:     if  $1 < m_q$  then  
9:       return  $x \cup y \cup \{q\}$   
10:    else  
11:      return  $x \cup y$   
12:    end if  
13:  end if  
14: end procedure
```

Tenga en cuenta que el primer llamado a esta función debe ser:

$\text{FuBaR_GtO}(M, 1, |M|)$

1. (10 %) ¿Qué hace el algoritmo FuBaR_GtO?

2. (10 %) Calcular el orden de complejidad del algoritmo `FuBaR_Gt0`:

2. Centro de gravedad de una secuencia

En una secuencia, el centro de gravedad *cog* se define como:

$$cog = \arg \min_{1 \leq k \leq |S|} \left| \sum_{i=1}^{k-1} s_i - \sum_{j=k+1}^{|S|} s_j \right|$$

el elemento que separa la secuencia en dos subsecuencias donde la diferencia absoluta entre las sumas de ellas es mínima.
En este punto usted debe escribir formalmente un algoritmo, basado en la estrategia "dividir-y-vencer", para calcular el centro de gravedad de una secuencia dada.

2.1. (30 %) Análisis del problema

2.2. (20 %) Diseño del problema

2.3. (10 %) Seudo-código

2.4. (20 %) Análisis de complejidad

2.5. (2 %) Invariante