Resolución de Tarea 1 - Complejidad Algorítmica y Análisis Amortizado (Fecha: 29 de Septiembre de 2025)

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5651 - Diseño de Algoritmos I
Septiembre - Diciembre 2025
Estudiante: Junior Miguel Lara Torres (17-10303)

Tarea 1 (9 puntos)

Indice

Pregunta 1

Consideraciones

Teniendo en cuenta los siguientes puntos

- $T_{permutaciones} \in \Theta(1)$
- $T_{ordenado} \in \Theta(n)$
- El arreglo **a** puede tener la propiedad de:
  - Un **multiconjunto** teniendo así una cantidad de permutaciones igual a
$$\binom{n}{m_1, m_2, ..., m_k} = \frac{n!}{m_1!m_2!...m_k!}$$
  - Un **conjunto** teniendo asi una cantidad de permutaciones igual a $n!$
- Para el punto 3, se tiene que, en cualquier caso, se cumple que:

$$\frac{n!}{m_1!m_2!...m_k!} \leq n!$$

Esto porque, cuando no hay repetición de elementos (conjunto), entonces se cumple $(\forall k | 1 \leq k \leq n : m_k = 1)$, por lo tanto:

$$\frac{n!}{1!1!...1!} \leq n! \iff n! \leq n!$$

En caso de haber al menos una repetición de un elemento, entonces:

$$\frac{n!}{m_1!m_2!...m_k!} < n!$$

Peor caso

Con el arreglo **a** teniendo propiedades de conjunto, tendremos $n!$ permutaciones para revisar. Adicionalmente, el predicado **ordenado** tarda $O(n)$, se tiene entonces $f = n! \cdot n)$, por lo tanto, $bogoMin \in O(n! \cdot n)$.

Mejor caso

Teniendo un arreglo **a** con propiedades de conjunto y la primera permutación sea la ordenada, y teniendo en cuenta que **ordenado** tarda $O(n)$, entonces al final tendríamos $g = n \cdot 1$, por lo tanto $bogoMin \in O(n)$.

Ya en caso de un arreglo con propiedades de multiconjunto, el mejor caso sería tomar la primera permutación que esté ordenada, así que se cumple el mismo análisis.

Pregunta 2

Siendo N el número de personas en la fila F. Como se habla del costo amortizado para cada **llamado** de $sombrerear()$ se debe tener encuenta que c_i se define como el **costo real** de la i-ésima llamada. Este costo es igual al numero de acciones de poner/quitar el sombrero. Sabiendo que la acción de poner/quitar un sombrero es $O(1) y como el llamado se detiene en el momento de **poner** un sombrero, podemos decir que $c_i = m$ donde m es la cantidad de sombreros totales cambiados en el **llamado**.

Así, se define la función de transferencia $\Phi(F)$ asociado al estado de la fila F, como el número total de personas que tienen el sombrero **puesto**.

$$\Phi = \text{Número de personas con el sombrero puesto}$$

Entonces, el costo amortizado de cada llamado de $sombrerear()$ se define como

$$Amor_{c_i} = c_i + \Phi(F_i) - \Phi(F_{i-1})$$

Analizamos por casos

- Caso 1: Se tienen m cambios con $m < N$

  Esto es que $m-1$ personas en la fila se quitaron el sombrero y la k-ésima se lo colocó. De esto obtenemos que

  - Costo real: $c_i = m$ por haber m cambios.
  - Cambio potencial:
    * Estado $i-1$: Se quitan $m-1$ sombreros porque habian $m-1$ sombreros puestos.
    * Estado i: Se coloca 1 sombrero.

  Sustituyendo tenemos que

  $$Amor_{c_i} = m + 1 - (m-1) = 2$$

- Caso 2: Se tienen m cambios con $m = N$

  Esto es que todas las personas tenien el sombrero puesto y se lo quitaron.

  - Costo real: $c_i = m = N$ por haber $m = N$ cambios.
  - Cambio potencial:
    * Estado $i-1$: Se quitan $m = N$ sombreros porque habian $m = N$ sombreros puestos.
    * Estado i: 0 porque no se coloca ningun sombrero.

  Sustituyendo tenemos que

  $$Amor_{c_i} = N + 0 - N = 0$$

Notamos que

$$Amor_{c_i} = \begin{cases} 2 & \text{si la operación se detiene en algún punto (Caso 1)} \\ 0 & \text{si la operación recorre toda la fila (Caso 2)} \end{cases}$$

Dado que $2 \in O(1) \wedge 0 \in O(1)$, entonces orden amortizado para cada llamado de sombrerear() es $O(1)$.

Pregunta 3