Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI5651 – Diseño de Algoritmos I
Enero–Marzo 2024

Tarea 1

A continuación encontrará 3 preguntas, cada una dirá cuántos puntos vale en su preámbulo. Sea lo más detallado y preciso posible en sus razonamientos, algoritmos y demostraciones.

La entrega se realizará <u>únicamente</u> por correo electrónico a rmonascal@gmail.com.

Fecha de entrega: Hasta las 11:59pm. VET del **Lunes, 22 de Enero** (*Semana 2*).

1. (3 puntos) — Considere el algoritmo `StrupidSort` para ordenar un arreglo de menor a mayor:

```
def stupidSort(a):
   for x in permutaciones(a):
      if ordenado(x):
         return x
```

   Puede suponer que `permutaciones` es un iterador que devuelve, una a una, todas las permutaciones de un arreglo en tiempo constante. Además, `ordenado` es un predicado que verifica si un arreglo está ordenado de menor a mayor en tiempo $\Theta(n)$.

   Diga cotas inferior y superior para `stupidSort` (correspondiendo al peor y mejor caso, respectivamente). Esto es, defina dos funciones f y g tal que se cumpla que:

   - `stupidSort` $\in O(f)$
   - `stupidSort` $\in \Omega(g)$

2. (3 puntos) — En la sede del banco caben hasta C personas. La sede empieza vacía y los clientes entran uno a uno. Cuando la capacidad de la sede es superada, la misma cierra y la totalidad de los clientes es transferida a una sede más grande, con capacidad $2 \times C$.

   Nótese que si contamos cuántas personas deben trasladarse al entrar un cliente a la sede, ocurre uno de dos casos:

   - Si aún no se ha superado la capacidad de la sede, la cantidad de personas que debe trasladarse es $O(1)$ — únicamente el cliente que entra.
   - Si se supera la capacidad de la sede, la cantidad de personas que debe trasladarse es $O(C)$ — todos los clientes en la sede del banco más el cliente nuevo.

   Demuestre que el orden amortizado, en el peor caso, para el ingreso de un cliente es $O(1)$

3. (3 puntos) — El problema $\oplus$–SAT modificación sobre el problema de satisfacibilidad booleana, donde la fórmula proposicional está en forma normal conjuntiva, pero cada cláusula está separada por disyunciones exclusivas (en vez de disyunciones tradicionales).

   Recordemos que $P \oplus Q$ es *cierto* sí y solo sí uno de entre P y Q es *cierto* (el otro siendo *falso*).

   Diga si $\oplus$–SAT $\in P$ o si $\oplus$–SAT es NP–completo. Demuestre su afirmación.

R. Monascal / Enero 2024