

# 1er Recuperatorio

Algoritmos y Estructuras de Datos 3 – DC, FCEyN, UBA

25/06/2021

Para realizar consultas, deben conectarse por Discord al canal de le docente a le cual quieran consultar. Tener en cuenta que un docente no puede conectarse con dos estudiantes en simultáneo. Las aclaraciones de enunciado que podamos llegar a hacer van a ser comunicadas vía Discord al canal de consultas de la práctica.

El examen transcurre de 17:00 a 21:00 hs. A las 21:00 se desconectarán los docentes de sus canales de Discord y tendrán hasta las 21:30 para realizar la entrega vía campus. El archivo subido al campus puede sobrescribirse una cantidad ilimitada de veces hasta la hora de entrega. Independientemente de si sobrescriben o no, deberán confirmar su entrega definitiva (que ya no podrá sobrescribirse). Sólo en caso de que el Campus estuviera saturado y no funcionara, sería adecuado realizar la entrega por mail a [algo3-doc@dc.uba.ar](mailto:algo3-doc@dc.uba.ar) con copia a [fsouthern@dc.uba.ar](mailto:fsouthern@dc.uba.ar) indicando claramente la entrega en el asunto.

El examen debe **realizarse a mano**. Deben **numerar sus hojas** y escribir en ellas sus **nombres y número de legajo** (i.e., libreta o DNI). Al finalizar, deben **escanearlo o fotografiarlo** y deben **unir y comprimir** las páginas resultantes para generar un único archivo en **formato PDF** con un **peso razonable**. El resultado debe ser un documento **legible** (buena iluminación, buena resolución, buena orientación, no fotos cortadas, etc.), **¡verificarlo!**. El archivo debe estar nombrado **apellido\_nombre.extensión** y debe haber un **orden de lectura claro**.

El examen es personal y pueden usar las teóricas, las clases prácticas y las guías de ejercicios, citando claramente. Las respuestas deben estar debidamente justificadas incluso en aquellos ejercicios en los que este hecho no es recordado.

El examen se **aprueba** con al menos 2 ejercicios bien.

- 1) GASTAVIDAS era un juego muy popular entre speedrunners hasta que Carle mostró que se puede ganar de forma óptima siguiendo una estrategia bien definida, lo que le restó interés al juego. Como resultado, los desarrolladores de GASTAVIDAS programaron GASTAVIDAS 2 con la esperanza de desafiar nuevamente a los jugadores. Al igual que su predecesor, GASTAVIDAS 2 es un juego con  $n$  niveles que deben ser pasados uno después del otro. Esta vez, sin embargo, los jugadores empiezan con un cantidad de vidas  $v_0$  potencialmente mayor a 1. Nuevamente, en cada nivel  $i$  se pueden conseguir hasta  $v_i$  vidas, para algún  $0 \leq v_i \leq n$ . Al igual que en la primer versión, la ventaja de conseguir muchas vidas es que las mismas pueden gastarse en los distintos niveles para acelerar la resolución del juego. Por ejemplo, si un nivel tiene un piso de espinas, el jugador puede pasar por encima de ellas en unos pocos segundos perdiendo una vida, o puede buscar un camino alternativo y más largo

sin gastar vidas. Obviamente, como en la versión 1, las vidas solo se pueden gastar luego de ser conseguidas y el juego termina una vez que le jugador se queda sin vidas. En GASTAVIDAS 2, la categoría *Beat Carle* tiene tres reglas particulares: 1. cada jugador debe recolectar todas las vidas de todos los niveles, 2. las vidas recolectadas en el nivel  $i$  no se pueden gastar en el nivel  $i$  y, 3. en cada nivel se pueden gastar a lo sumo 100 vidas. Luego de un arduo entrenamiento, Panche ha logrado resolver cada nivel  $i$  en tiempo  $t_{ip} \in \mathbb{N}$  cuando gasta  $p$  vidas, para  $0 \leq p \leq 100$ . A diferencia de Carle y, debido a los desafíos que impone la versión 2 en su diseño de niveles, ya **no es cierto** que  $t_{ip} \geq t_{ip+p'}$  para todo  $0 \leq p \leq p + p' \leq 100$ . Igualmente, Panche quiere lograr superar a Carle y por lo tanto pretende determinar la estrategia óptima para GASTAVIDAS 2.

- a) Definir en forma recursiva la función  $G: \{1, \dots, n\} \times \mathbb{N} \rightarrow \mathbb{N}$  tal que  $G(i, v)$  denota la mínima cantidad de tiempo que le toma a Panche resolver los niveles  $i, \dots, n$  si empieza con  $v$  vidas.
  - b) Demostrar que  $G$  tiene la propiedad de superposición de subproblemas.
  - c) Definir un algoritmo *top-down* para calcular  $G(i, v)$  cuando  $V = \{v_i \mid 1 \leq i \leq n\}$  y  $T = \{t_{ip} \mid 1 \leq i \leq n, 0 \leq p \leq 100\}$  son dados como input, indicando claramente las estructuras de datos utilizadas y la complejidad resultante. Nota: para que el ejercicio se considere bien es necesario que la complejidad temporal del algoritmo sea  $O(n^2)$ .
  - d) Escribir el (pseudo-)código del algoritmo top-down resultante.
- 2) Dado un digrafo  $G$  y tres vértices  $s, r$  y  $t$ , queremos encontrar un recorrido de  $s$  a  $t$  que pase por  $r$ , tenga longitud par y use la menor cantidad de aristas. Diseñar un digrafo  $H$  con dos vértices  $r_1$  y  $r_2$  de forma tal que la respuesta al problema se pueda obtener con dos invocaciones de BFS (consultando distancias particulares en los árboles resultantes), una desde  $r_1$  y otra desde  $r_2$ . **Justificar** que el algoritmo propuesto resuelve el problema.
  - 3) Decimos que un grafo  $G$  es un *cactus* cuando cada una de sus aristas pertenece a lo sumo a un ciclo del grafo. Diseñar un algoritmo que encuentre el ciclo de longitud máxima cuando un cactus  $G$  es dado. El algoritmo debe tener una complejidad temporal  $O(n)$ . **Justificar** que el algoritmo propuesto es correcto. **Nota:** recordar que todos los cactus tienen  $O(n)$  aristas, tal como se vio en el primer parcial; no hace falta demostrar este hecho.
  - 4) Diseñar un algoritmo eficiente que dado un digrafo pesado  $G$  y dos vértices  $s$  y  $t$ , determine el recorrido mínimo de  $s$  a  $t$  que pasa por a lo sumo una arista de peso negativo. **Justificar** que el algoritmo propuesto es correcto.