Parcial de Programación 3
Simulacro

- Este parcial dura **3** horas y consta de 1 carillas. El total de puntos es **50**.
- **NO** se puede utilizar ningún tipo de material de consulta. Salvo que se indique lo contrario podrá usarse todo lo visto en el teórico, práctico y laboratorio sin demostrarlo, indicando claramente lo que se está usando.

Se requiere:

I) Numerar todas las hojas e incluir en cada una el **nombre, cédula de identidad, número de página y cantidad de hojas entregadas**.

II) Utilizar las hojas de **un solo lado** y escribir con lápiz.

III) Iniciar cada ejercicio en hoja nueva.

Ejercicio 1 (15 puntos)

Sea $G = (V, E)$ un grafo (no dirigido) conexo con n vértices y m aristas.

(a) Escriba un algoritmo que encuentre un ciclo de largo mínimo en G o reporte que G es acíclico. Su algoritmo debe admitir una implementación cuyo tiempo de ejecución sea $O(mn)$. Reescriba cualquier algoritmo que use de los estudiados durante el curso.

(b) Demuestre que el tiempo de ejecución de su algoritmo es $O(mn)$. Repita cualquier argumento que use del material teórico del curso.

Sugerencia: Ejecute BFS desde cada uno de los nodos de G.

Ejercicio 2 (18 puntos)

Un centro de cómputo dispone de una supercomputadora y n servidores exactamente iguales. Necesita ejecutar n procesos, cada uno de los cuales consta de 2 etapas: una etapa de preprocesamiento que se ejecuta en la supercomputadora en primer lugar y una etapa de posprocesamiento que se realiza a continuación en alguno de los servidores. Cada proceso i, $1 \leq i \leq n$, requiere un tiempo a_i de ejecución en la supercomputadora y un tiempo b_i de posprocesamiento en algún servidor. La supercomputadora puede ejecutar un proceso a la vez.

Queremos minimizar el tiempo total de procesamiento, es decir, el tiempo transcurrido desde que se inicia la ejecución del primer proceso en la supercmputadora hasta que termina la etapa de posprocesamiento del proceso que termina más tarde en un servidor.

(a) Dé un algoritmo eficiente que determina un orden de ejecución de las tareas para resolver el problema de minimización planteado.

(b) Demuestre la corrección de su algoritmo.

Ejercicio 3 (17 puntos)

Tenemos un arreglo A con n números distintos ordenados de menor a mayor, donde n es una potencia de 2 mayor o igual a 2. Por otra parte tenemos otro arreglo, B, que contiene todos los números de A salvo uno que fue eliminado. Nos interesa saber cuál es el elemento faltante, pero el arreglo B **no está ordenado**.

(A) Escriba un algoritmo que sigue una estrategia de tipo divide y vencerás para encontrar el elemento faltante en tiempo $O(n)$.

(B) Justifique que el tiempo de ejecución es $O(n)$ escribiendo una relación de recurrencia y explicando el origen de cada término (no es necesario resolverla).

Sugerencia: Use una estrategia similar a la que emplea QuickSort para dividir el problema original.