

Este es el primer examen parcial del curso *Análisis y Diseño de Algoritmos*, 2020-2. El examen tiene 3 preguntas; otorga un total de 50 puntos y 20 de bono. El examen es *individual* y no es permitido discutir sobre su contenido (e.g., enunciados, estrategias de solución, soluciones) con nadie más.

Nombre y código: \_\_\_\_\_

Pregunta	1	2	3	Total
Puntos	20	15	15	50
Bono	10	0	10	20
Puntaje				

- Con base en el enunciado del problema *Local minimums* (Ejercicio 1.32, página 61 de *Algorithms* por J. Erickson):
  - (5 puntos) Especifique el problema dado.
  - (5 puntos) Usando la técnica de dividir y conquistar, diseñe un algoritmo que en tiempo  $O(\log N)$  resuelva el problema dado, en donde  $N$  es el tamaño del arreglo de entrada.
  - (10 puntos) Demuestre que el algoritmo es correcto con respecto a la especificación usando inducción matemática.
  - (10 +) Implemente el algoritmo en el lenguaje de programación Python. *Nota: este bono solo es acumulable si el algoritmo es correcto.*
- Con base en el enunciado del problema *The big dance contest* (Ejercicio 3.13, página 131 de *Algorithms* por J. Erickson) y la metodología propuesta en el curso para resolver problemas usando la técnica de programación dinámica:
  - (5 puntos) Plantee una función recurrente que permita resolver el problema, enunciando claramente la descripción de la función, restricciones sobre sus parámetros y cómo se puede usar para resolver el problema dado (es necesario partir de una especificación del problema).
  - (10 puntos) Determine si es más conveniente utilizar memorización o tabulación, justificando su respuesta, y diseñe un algoritmo (incluyendo invariantes) en coherencia con su decisión y justificación.
- Con base en el enunciado del problema *High-performance computing* (Ejercicio 6.9, página 320 de *Algorithm Design* por J. Kleinberg y E. Tardos):
  - (5 puntos) Resuelva el numeral (a) del problema.
  - (10 puntos) Resuelva el numeral (b) del problema usando la metodología propuesta en el curso y analizando la complejidad asintótica de su algoritmo (tiempo y espacio). Justifique sus decisiones de diseño.
  - (10 +) Diseñe un algoritmo que utilice la técnica de tabulación y minimice la cantidad de espacio necesario para ella, indicando cuál es su complejidad asintótica (tiempo y espacio).