

# Primer Parcial de Algoritmos y Estructuras de Datos I

Recordá que el parcial es individual y debés resolverlo en papel con birome o lápiz; es importante que sea legible. En todas las páginas debe constar claramente tu nombre, apellido y firma. Para entregarlo tenés que enviar un solo mail con fotos de todas las páginas a las direcciones de email: miguel.pagano@unc.edu.ar, francolq@unc.edu.ar, demetriomeister@gmail.com, El asunto del mail debe ser: "Primer Parcial de AYED1".

Sólo se considerará entregado si el mail tiene hora de salida antes de las 18.00 (hora Argentina) del martes 27 de octubre.<sup>1</sup> Si tenés dudas, podés plantearlas en la sala del teórico el martes desde las 9.00 hasta las 13.00.

1. Una forma de *medir* cuán distintas son dos listas de la misma longitud es contando la cantidad de posiciones en los que las palabras difieren. A esa medida se la conoce como la distancia de Hamming; nuestro objetivo es especificar y definir la función *ham*. Algunos ejemplos:

1. *ham*."sopa"."sapo" = 2.
2. *ham*."pedro"."perro" = 1
3. *ham*."pan"."queso" = 3.

La especificación de *ham* es la siguiente:

$$ham.xs.ys = \langle \exists i : 0 \leq i < \#xs \wedge \#ys : xs[i] \neq ys[i] \rangle$$

- a. (0.5 pt) Determiná el tipo de *ham*.
- b. (0.5 pt) Explicá con *tus propias palabras* cómo harías la derivación de *ham*.
- c. (0.5 pt) Encontrá cuánto vale *ham*."come"."toma".
- d. (0.5 pt) Derivá el caso base.
- e. (1 pt) Derivá el caso inductivo.
- f. (1 pt) Hacé la evaluación (paso-a-paso) de *ham*."toma"."come".

2. En este ejercicio nos preguntamos cuándo una lista de números se pueden partir en dos segmentos de manera tal que sus sumas sean iguales:<sup>2</sup>

$$pj.xs = \langle \exists as, bs : xs = as ++ bs : sum.as = sum.bs \rangle$$

- a. (0.5 pt) ¿Te parece que para derivar *pj* tenés que derivar *sum*? Justificá tu respuesta con tus propias palabras.
- b. (0.5 pt) Proponé *ns* tal que  $\#ns = 2$  y *pj.ns* = *False*. Demostrá que efectivamente se cumple la segunda condición, evaluando la especificación en la lista que diste.
- c. (0.5 pt) Proponé *ms* tal que  $\#ms \geq 3$  y *pj.ms* = *True*. Explicá con tus propias palabras por qué se satisface la segunda condición.
- d. (0.5 pt) Derivá el caso inductivo de *pj* hasta donde puedas, indicando claramente la hipótesis inductiva.
- e. (0.5 pt) Proponé la generalización necesaria para continuar con la derivación.
- f. (0.5 pt) Derivá el caso base de la función generalizada.
- g. (1 pt) Derivá el caso inductivo de la función generalizada.

3. En este último ejercicio nos interesa explorar cómo especificar formalmente problemas sencillos que luego querríamos derivar. Si necesitás modularizar la especificación está bien que lo hagas.

1. La lista *es* de números enteros tiene un prefijo creciente y un sufijo decreciente.
  2. Ningún elemento de la lista *as* satisface el predicado *p*.
  3. Sumatoria de los primos menores que *k*.
- a. (1pt) Elegí una (sí, sólo una) de las frases y proponé una especificación formal para esa frase. Para eso, dale un nombre a la función, indicá el tipo y escribí tu especificación (como hicimos más arriba con *ham* y *pj*) indicando también los parámetros.
  - b. (1pt) Justificá con tus palabras por qué esa especificación formal es correcta.

<sup>1</sup>Esta hora la determina gmail y no tu compu/celular/tablet.

<sup>2</sup>Podríamos decir que la lista se puede partir de manera justa, de ahí el nombre: *pj*.