

# Analisis de algoritmos 2022-1

## Tarea 1

Fecha de entrega: viernes 8 de octubre de 2021

### 1 Ejercicios

Considera el problema MIN: dado un arreglo  $A$  de  $n$  elementos comparables, devolver el valor del elemento más pequeño de  $A$ .

1. Diseña un algoritmo iterativo que resuelva el problema MIN en tiempo  $O(n)$ . Demuestra usando invariantes que es correcto. Demuestra que su complejidad es realmente  $O(n)$ .
2. Diseña un algoritmo recursivo que resuelva el problema MIN en tiempo  $O(n)$ . Demuestra por inducción en el valor de  $n$  que es correcto. Demuestra que su complejidad es realmente  $O(n)$ .
3. Diseña un algoritmo que en tiempo lineal calcule la profundidad de un árbol binario  $T$  dado. La profundidad de un árbol se define como el máximo número de aristas que separan la raíz con alguna hoja del árbol. Demuestra que el algoritmo es correcto, y que su complejidad es lineal en el número de vértices y aristas de  $T$ .
4. Diseña un algoritmo que, dado un arreglo  $A$  de  $n$  enteros, y un entero  $x$ , devuelva el índice más pequeño tal que  $A[i] == x$ , o el valor -1 si  $x$  no aparece en el arreglo. Propón el invariante natural de tu algoritmo, demuéstralo, y úsalo para concluir que el algoritmo es correcto. Demuestra porqué es de tiempo  $O(n)$ .

Ejercicio opcional. Demuestra el siguiente lema (sugerimos usar inducción): Todo árbol de al menos dos vértices, tiene al menos dos hojas (una hoja es un vértice de grado 1).

Con el uso de ese lema, diseña un algoritmo que, dado un árbol de  $n$  vértices, coloreé todos los vértices del árbol con colores 0 y 1, de tal forma que dos vértices adyacentes tengan colores distintos. Demuestra que el

algoritmo es correcto. En este ejercicio no hace falta demostrar complejidad. Expresa el algoritmo en el mayor nivel de abstracción posible: no te preocupes por la representación en estructura de datos del árbol.

## 2 Rúbrica de evaluación

Consideren los siguientes puntos a revisar para la elaboración de su tarea.

1. Para todos los ejercicios tener presente:
  - El algoritmo está bien planteado.
  - La prueba contempla todos los detalles.
  - La prueba está libre de detalles innecesarios.
  - El lenguaje utilizado es simple y claro.
2. Para los ejercicios que requieren el uso de invariante tener presente:
  - El invariante está bien planteado.
  - El argumento del porqué el invariante implica la corrección del algoritmo es correcto y completo.
3. Para los ejercicios que requieren el uso inducción matemática tener presente:
  - El caso base está bien planteado.
  - La hipótesis de inducción está bien planteada.
  - El caso inductivo está bien planteado.
  - El caso inductivo hace uso explícito de la H.I.