



Universidad Nacional del Comahue
Facultad de Informática
Departamento de Programación



Diseño de Algoritmos

Trabajo Práctico 0

Ignacio Alejandro Navarro Oliva

2024

Índice

1. Ejercicio 1	3
2. Ejercicio 2	3
2.1. a) Insertar un elemento en la posición k de una lista	3
2.2. b) Poner un elemento en una cola	3
2.3. c) Buscar un elemento en un árbol binario	3
2.4. d) Eliminar un elemento en un árbol binario de búsqueda	3
2.5. e) Insertar un elemento en una tabla Hash	3
3. Ejercicio 4	4
3.1. Un juego de ajedrez.	4
3.2. Un determinado problema.	4

1. Ejercicio 1

Establecer el orden de las siguientes fracciones de código:

- Dos estructuras *for* anidadas más una condicional *if-else* con tiempos constantes en ambos cuerpos: corresponde a un orden cuadrático $O(n^2)$
- Una estructura *while* y operaciones constantes: corresponden a $O(n)$

2. Ejercicio 2

Decir cuál es el orden de las siguientes operaciones y justificar:

2.1. a) Insertar un elemento en la posición k de una lista

Teniendo en cuenta la implementación de una lista común, se desplaza desde la primera posición de la lista hasta la posición k e inserta el elemento. Por lo tanto, toma k pasos (es decir, $O(n)$).

2.2. b) Poner un elemento en una cola

Tiene tiempo constante ($O(1)$), ya que se coloca desde el frente (no hay que recorrer nada).

2.3. c) Buscar un elemento en un árbol binario

Un árbol binario es un árbol en el que cada nodo tiene como máximo 2 hijos. La operación de inserción toma distinto tiempo para un mismo elemento dependiendo de cómo recorre el árbol, pero en general el peor caso es $O(n)$. No hay ningún recorrido en el árbol que sugiera la posibilidad de un menor tiempo.

2.4. d) Eliminar un elemento en un árbol binario de búsqueda

En contraste con el punto anterior, en un árbol binario de búsqueda encontrar un elemento toma tiempo logarítmico, y eliminar el elemento no supone un aumento más que de una constante. $O(\log n)$.

2.5. e) Insertar un elemento en una tabla Hash

Tiempo constante $O(1)$.

3. Ejercicio 4

3.1. Un juego de ajedrez...

Un juego de ajedrez se puede plantear como el siguiente problema de decisión: dado un posicionamiento legal de las piezas de ajedrez y la información sobre qué mueve, determinar si ese lado puede ganar, ¿es decidible?

Sí, ya que por fuerza bruta se podrían observar todas las combinaciones posibles y obtener alguna secuencia de movimientos en las cuales el juego termine en victoria (esto si interpretamos el enunciado diciendo, ¿es decidible determinar si ese lado puede ganar después de *un* solo movimiento?). Claramente, no sería eficiente.

3.2. Un determinado problema...

Un determinado problema puede ser resuelto por un algoritmo cuyo tiempo de ejecución está en $O(n \log n)$. ¿Cuál de las siguientes afirmaciones es verdadera?

- el problema es manejable
- el problema es intratable
- imposible decirlo

Es posible afirmar que el problema es manejable. Tiene una solución con tiempo polinómico, por lo que en cuanto a ese recurso se puede afirmar que está en P . En cuanto a espacio, siempre más o menos se condice con el tiempo que lleva el algoritmo, por lo que no cambiaría mucho en ese aspecto.