Análisis de Algoritmos 2024

Trabajo Práctico 4.1 - Introducción a Recurrencias
21 de octubre de 2024

21 de octubre de 2024

El objetivo de este trabajo es que realicemos colaborativamente los ejercicios propuestos. Para los ejercicios compartidos entre grupos, podés compartir el desarrollo del ejercicio o realizar tu propia resolución y publicarla en este documento.
Debés resolver todos los ejercicios, pero en este trabajo sólo debés publicar los que te tocaron según la asignación de la tabla siguiente. Dicha asignación también aplica a la presentación oral del TP.
Por supuesto, si querés compartir tus soluciones de otros ejercicios que se te asignaron en este proyecto Overleaf, también son bienvenidas pero no es obligatorio.
Colocá tu nombre y apellido en el ejercicio que te tocó al iniciar la resolución del ejercicio que te tocó.
Resolvé el ejercicio que te tocó a continuación del enunciado.

Para publicar tus soluciones de código escribí, en este tp según corresponda en cada ejercicio, el link (con formato link) a tu repositorio público asociado.

Grupos (Estudiantes por grupo)	Ejercicio	Ejercicio	Ejercicio
G1 (Sthefany, Victoria, Paula, Adriano, Facundo)	1.1	2.1.i	2.1.b
G2 (Sebastián, Bautista, Antonio, Albany, Luis)	1.2	2.1.g	2.1.c
G3 (Lucas, Valentina, Nicanor, Ignacio, Franco)	1.1	2.1.h	2.1.a
G4 (Manuel, Jan, Ulises, Demian , Luciano)	1.2	2.1.i	2.1.b
G6 (Roman, Ulises, Guillermo, Jamiro, Juan)	1.1	2.2	2.1.a
G7 (Christopher, Tomás, Joaquín, Leonard, Facundo)	1.2	2.1.f	2.1.b
G8 (Facundo, Belén, Jeremías, Bruno, Kevin)	1.1	2.2	2.1.c

1. Soluciones Iterativas vs Recursivas

1. Considerá el problema de hallar el n-ésimo número de la secuencia de Fibonacci.

   a) Implementá una solución iterativa y una recursiva en Java. Utilizá tu entorno preferido para implementar tu solución. Comparti dicho código de algún modo (de preferencia github).

   b) Compará empíricamente los tiempos de ejecución de estos algoritmos (iterativo y recursivo). Para ello, registrá el tiempo de ejecución de ambas soluciones en nanosegudos (ns) o milisegundos (ms), probá varios casos y hacé una tabla comparativa. Podés hacer un programa de test para preconfigurar las pruebas. También podés crear un archivo de salida con el tiempo registrado de cada ejecución.

   c) Realizando como mínimo 10 ejecuciones calculá la media aritmética ($\overline{x}$), el error absoluto de cada una y el error promedio ($E_{\overline{x}}$).

d) Calculá la función teórica $T(n)$ de tiempo de ejecución de la solución iterativa. ¿El tiempo calculado tiene relación con el tiempo empírico registrado en el punto anterior?

e) Predecí los tiempos de ejecución teóricos (en ms o ns) utilizando y $E_{\overline{x}}$.

f) Escribí la función recurrente de la solución recursiva. La retomaremos el el siguiente práctico para resolverla y compararla con la solución iterativa y las pruebas empíricas.

g) ¿Qué pasa con la solución recursiva para un Fibonacci muy grande? ¿Es posible obtener el tiempo empírico?

2. Recuperá del TP2.2: Omicron, Theta, Omega la solución iterativa y recursiva lograda del ejercicio 3.2. Para recordar su enunciado:

   Implementar un algoritmo tal que dado un número entero n (con $n < 10^{18}$) compute la parte entera de la raíz cuadrada de n, usando solo operaciones primitivas $(+, , , /, =, ==, <, >)$ y cualquier estructura de control.

   a) Recuperá las implementaciones de ambas soluciones (iterativa y recursiva).

   b) Compará empíricamente los tiempos de ambas soluciones. Realizá un registro de la compartiva como en el ejercicio anterior.

   c) Realizando como mínimo 10 ejecuciones calcular la media aritmética ($\overline{x}$), el error absoluto de cada una y el error promedio ($E_{\overline{x}}$).

   d) Recuperá la función teórica $T(n)$ de tiempo de ejecución para la solución iterativa. Si realizaste mejoras en el punto a) anterior, corregí el tiempo de ejecución obtenido. ¿El tiempo calculado tiene relación con el tiempo empírico registrado en el punto anterior?

   e) Escribí la recurrencia asociada a la solución recursiva.

2. Escribiendo recurrencias

1. Accede a la siguiente página: `https://projecteuler.net/archives`[1]. Allí encontrarás una lista de situaciones problemáticas. Podés resolver cualesquiera de ellas pero en este ejercicio te pedimos que trates específicamente las listadas a continuación.

   - Resolvé de forma recursiva los siguientes problemas
   - Expresá la relación de recurrencia asociada a la solución recursiva obtenida.

   a) Problema 1: Múltiplos de 3 y 5.

   b) Problema 2: Suma de números pares de Fibonacci.

   c) Problema 3: El factor primo más grande.

   d) Problema 4: Producto palíndromo más grande.

[1] **Project Euler.net** recopila problemas matemáticos orientados a programación que requieren un poco más que sólo conocimientos matemáticos. Se distrubuyen bajo licencia Creative Commons: *Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)*

e) Problema 7: El número primo 10001-ésimo.

f) Problema 10: Suma de números primos.

g) Problema 15: El camino enredado.

h) Problema 17: Cuenta las letras de los números.

i) Problema 18: La suma del camino máximo.

2. Sea a_n el número de palabras de longitud n formado con $0s$ y $1s$ sin $0s$ consecutivos. Encontrá una relación de recurrencia para el algoritmo recursivo que resuelve este problema.