CC3001 Algoritmos y Estructuras de Datos
Profesores: Nelson Baloian, Patricio Poblete, e Iván Sipirán
Auxiliares: Valentina Alarcón Yáñez, Samuel Chávez Fierro, Antonia G. Calvo, Cristián Llull, y Raimundo Lorca Correa

Auxiliar 2

28 de marzo de 2025

P1. Contando ocurrencias

El DCC estaba a punto de finalizar uno de los problemas del milenio, pero justo cuando estaban por contar ocurrencias en una lista de largo `n` ¡Se nos rompieron todos los ciclos! y ya no tenemos acceso a `for, while, do while` (¡Ni a ChatGPT!)

Por eso el departamento llama a la nueva estrella de programación (¡Usted!) que los rumores dicen que aprendió una técnica oscura y le piden si puede encontrar una forma alterna de solucionar este dilema.

(a) Plantee el problema de forma usual, con ciclos y concluya que puede ser resuelto de forma recursiva

(b) Encuentre el caso base

(c) Programe la función `contarOcurrencias(array, elemento)`

P2. Triángulo de Pascal

En cuanto usted terminó de contar recurrencias, resulta que el DIM también estaba a punto de finalizar otro problema del milenio, y justamente cuando lo que les faltaba era armar un triángulo de Pascal, también perdieron nuevamente los ciclos

El DIM escuchando los rumores del DCC, se entera de la nueva estrella que puede resolver problemas iterativos con esta técnica que llamó "recursión"

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

Triángulo de Pascal de referencia

Una propiedad interesante del triángulo de Pascal es que en la i-ésima fila (contando desde 0), la j-ésima posición (nuevamente contando desde 0) contiene el valor del binomial $\binom{i}{j}$

La idea del ejercicio es usar como caso recursivo la regla de Pascal:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

(a) Plantee el problema de forma usual, con ciclos y vea los paralelismos de como puede ser recursivo

(b) Encuentre el caso base

(c) Programe la función `trianguloPascal(n)` que imprime los n primeros niveles del triángulo de pascal

P3. ¿Se acuerdan de Pokémon™?

En el auxiliar 0, tuvimos la dicha de ser un entrenador Pokémon. Sin embargo, si alguno conoce el juego, no se tiene la experiencia completa si no aparecen ¡enemigos!

Ahora vamos a modelar a este enemigo con diagramas de estado.

(a) Modelamiento de la batalla

Modele una batalla entre el enemigo y el entrenador Pokémon. *Recordatorio: en una batalla, los Pokémon se atacan entre sí. Si se le acaba el HP, sigue batallando el siguiente Pokémon del entrenador. La batalla acaba cuando uno de los entrenadores se queda sin Pokémon.*

Defina estados y eventos y plantee el diagrama de estados.

(b) Modelando el comportamiento NPC

Modele el comportamiento NPC del enemigo. *Recordatorio: un enemigo llama a duelo a un entrenador Pokémon cuando se le cruza por enfrente. El entrenador no puede rechazar el desafío.*

Defina estados y eventos y plantee el diagrama de estados.

P4. Descomposición en factores primos (Propuesto)

Luego de un agotador día de maquinar problemas del milenio, usted decide relajarse, y se entera que los ciclos no solo se rompieron en la FCFM, ni en el país, sino que ¡Se rompieron de forma global!.

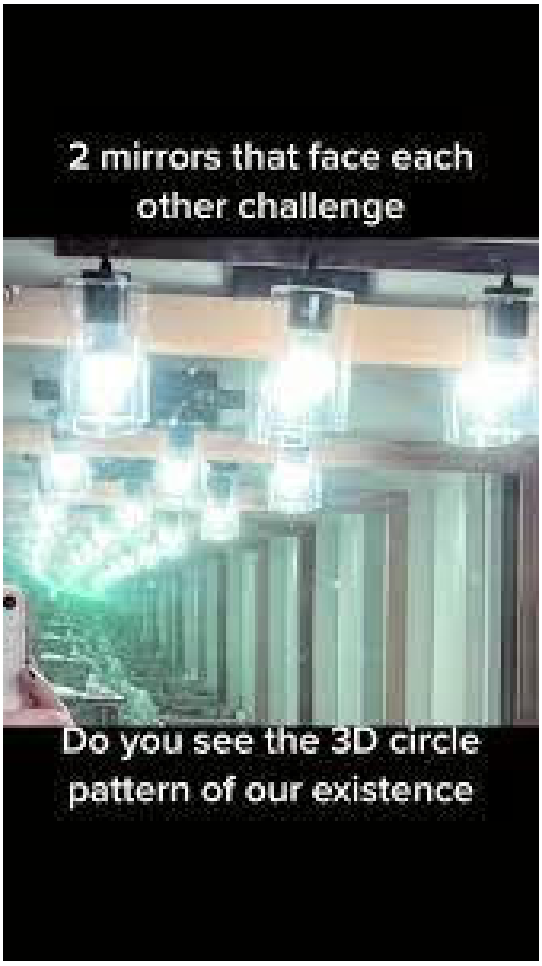
Google quienes estaban reparando la caída de iteraciones mundial, se encontró con un gran problema y necesita por $1T USD encontrar una forma de la factorización prima de un numero sin iteraciones que nos devuelva finalmente la iteración al mundo

$$2 \times 2 \times 3 \times 5 \times 7 = 420$$

Factorización prima de ejemplo

(a) Encuentre los casos base

(b) Programe la función recursiva `factoresPrimos(i, n)` que retorna una lista con todos los factores primos de n que son mayores que i.



To understand recursion, one must first understand recursion