

Olimpiada Informática Española

Soluciones de los problemas

2007

Las Torres de Hanoi [Ad-hoc]

Problema: En este problema se propone un algoritmo para resolver el problema de las torres de Hanoi con n discos. Dado un entero m se pide que se diga cuantos discos hay en cada palo, tras m pasos del algoritmo.

Solución: La posición del i -ésimo disco (empezando por 0) depende del valor $k = \frac{m}{2^i} \bmod 6$

- si $k = 5$ o $k = 0$ el disco está en el palo 0.
- si i es par y $k = 1$ o $k = 2$ el disco está en el palo 1.
- si i es par y $k = 3$ o $k = 4$ el disco está en el palo 2.
- si i es impar y $k = 1$ o $k = 2$ el disco está en el palo 2.
- si i es impar y $k = 3$ o $k = 4$ el disco está en el palo 1.

Como los números son muy grandes, se tiene que operar con strings.

Baldosas [Backtracking]

Problema: Dado un rectángulo $f \times c$ llenarlo con piezas $a \times b$, cada una correspondiente a una letra. Se pide la solución lexicográficamente menor.

Solución: La solución es un [backtracking](#). Pondremos las piezas por orden empezando desde la esquina superior izquierda y yendo horizontalmente. Siempre intentaremos poner primero las piezas de manera horizontal.

Casete [Greedy, Sorting]

Problema: Dado un conjunto de n canciones de duración d_i ; un casete con t segundos de duración, calcular el máximo número de canciones que se pueden guardar en el casete, si las canciones deben estar separadas por 1s

Solución: Ordenamos las canciones por duración, y vamos seleccionando las más cortas hasta que no podamos más.

Intérprete [Implementación]

Problema: Dado un programa en cierto lenguaje ensamblador, simular su ejecución. En el enunciado se dan las operaciones que puede tener y que representan.

Solución: Seguimos las instrucciones del enunciado y simulamos la ejecución.

Hojas y abejas [Implementación]

Problema: Dada un conjunto de plantas, podemos cortar una planta que tenga una flor por F euros y vender cada una de sus hojas por L euros. Podemos cortar flores solo si no tienen abejas.

Solución: Cortaremos flores que tengan n hojas solo si $n \cdot L > P$.

Paréntesis [Implementación]

Problema: dada una secuencia de paréntesis y claudators, indicar si está bien parentizada.

Solución: Simplemente recorreremos la secuencia viendo que nunca haya más cerrados que abiertos, y que al final el número de cerrados y abiertos sea igual (en cada caso). También debemos vigilar que no haya un paréntesis mal cerrados dentro de claudators y viceversa.

Sopa de letras [Implementación]

Problema: Contar cuantas veces aparece la palabra OIE en una sopa de letras.

Solución: Simplemente miramos en cada dirección cuantas veces aparece.

Permutaciones [Divide and conquer]

Problema: Se da una permutación expresada en ciclos y m preguntas. En cada una se pide la permutación después de ser aplicada sobre si misma k veces.

Solución: Primero transformamos la permutación en ciclos a una permutación normal p , sabiendo que $p[i]$ es el número que aparece en el ciclo de i justo después de ese número. Después, para calcular p^k usaremos el método de [exponenciación rápida](#).

Cambio mínimo [Greedy]

Problema: Dada una cantidad de dinero, pagarla usando el mínimo número de billetes/monedas.

Solución: La idea es la de un **algoritmo voraz**, cogemos en cada momento la moneda/billete más alto. Importante calcular las cantidades usando divisiones, para ir rápido.

Escape del laberinto [Implementación]

Problema: Dada una cuadrícula de $n \times m$ con obstáculos, calcular en número de maneras de ir desde la casilla inferior derecha a la superior izquierda. Solo nos podemos mover hacia la izquierda o hacia arriba.

Solución: La solución esperada consiste en una **dinámica**, los estados son $dp[i][j]$ que representa el número de maneras de llegar a la casilla (i, j) . La recurrencia sigue:

$$dp[i][j] = dp[i - 1][j] + dp[i][j - 1]$$

es decir, el número de maneras para llegar a la casilla (i, j) es la suma de las maneras de llegar a las casillas de las que puedo venir.

Dibujando rectángulos [Implementación]

Problema: Pintar un rectángulo como se muestra en los ejemplos del enunciado.

Solución: Realizamos un doble for, y dependiendo de los índices en los que nos hallemos, imprimimos un número u otro.

Campanadas en la Torrasa [Implementación]

Problema: Dada una hora, indicar el número de campanadas que hay que dar, o cuando será la próxima hora en que se den.

Solución: Seguimos las instrucciones que nos dan en el enunciado, donde se indica como calcular el número de campanadas dependiendo de la hora y los minutos.