

PRÁCTICA 6. Algoritmos Voraces. Programación Dinámica. Algoritmos vuelta atrás. Problema del cambio de monedas. Laberinto.**Nota:**

Junto con esta práctica se adjuntan varios archivos adicionales:

- 1) La clase abstracta EsquemaVZ desarrolla el esquema de resolución de algoritmos voraces.
- 2) La clase Laberinto.java, a falta de completar.
- 3) Los distintos juegos de prueba.

Ejercicios a realizar

- 1) Amplíe el ejercicio 3 de la práctica 4 que resuelve el problema del cambio de moneda recursivo con dos nuevas soluciones:
 - a) Escriba un algoritmo voraz rápido que resuelva el problema.
 - b) Escriba un programa que resuelva el problema con la estrategia de la programación dinámica.

Nota: Incluir en el paquete: **com.mp.practica6.ejercicio1**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica6.ejercicio1.test**

- 2) Escriba el algoritmo voraz que resuelva el problema del cambio de moneda utilizando la clase abstracta EsquemaVZ, esta clase se llamará CambioMonedaVoraz.

Nota: Incluir en el paquete: **com.mp.practica6.ejercicio2**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica6.ejercicio2.test**

- 3) Los algoritmos de vuelta atrás (también conocidos como de backtracking) hacen una búsqueda sistemática de todas las posibilidades, sin dejar ninguna por considerar. Cuando intenta una solución que no lleva a ningún sitio, retrocede deshaciendo el último paso, e intentando una nueva variante desde esa posición (es normalmente de naturaleza recursiva). El proceso general de los algoritmos de vuelta atrás se contempla como un método de prueba y búsqueda, que gradualmente construye tareas básicas y las inspecciona para determinar si conducen a la solución del problema. Si una tarea no conduce a la solución, prueba con otra tarea básica. Es una prueba sistemática hasta llegar a la solución, o bien determinar que no hay solución por haberse agotado todas las opciones que probar. La característica principal de los algoritmos de vuelta atrás es intentar realizar pasos que se acercan cada vez más a la solución completa. Cada paso es anotado, borrándose tal anotación si se determina que no conduce a la solución, esta acción constituye la vuelta atrás. Cuando se produce una vuelta atrás se ensaya otro paso (otro movimiento). En definitiva, se prueba sistemáticamente con todas las opciones posibles hasta encontrar una solución, o bien agotar todas las posibilidades sin llegar a la solución.

- a) Escriba un programa que resuelva un laberinto sencillo. Utilice parte del código que acompaña esta práctica. Señale los caminos de salida hacia la casilla marcada con "s" utilizando el carácter "x". La entrada en el laberinto debe ser una posición válida de la primera fila " ".

Nota: Incluir en el paquete: **com.mp.practica6.ejercicio3**

Nota: Las pruebas de unidad se situaran en el paquete : **com.mp.practica6.ejercicio3.test**