PRÁCTICA CUDA

JUEGO 16384

Laura Pérez Medeiro

Ángel Martín Segura

11 de abril de 2019

En este documento se especificarán los apartados completados de la práctica, explicando con mayor detalle algunos de los procedimientos más complejos que podemos encontrar en la práctica y se comentará la problemática que se ha tenido a la hora de elaborar algunos apartados.

1. Parte obligatoria

Para este apartado, las funciones que encontramos son:

* main(args: Array[String]): Función principal del programa, es la que inicia la ejecución, muestra el menú y llama a “jugar” en función de la dificultad introducida.
* tamTablero(dificultad:Int): (Int, Int) = dificultad match: Función encargada de asignar unas dimensiones al tablero, a raíz de la dificultad elegida.
* crearTablero(tam:Int):List[Int] =: Función encargada de crear un tablero vacío a raíz de la dificultad indicada.
* cogerLibres(tablero: List[Int]): List[Int] =: Se encarga de seleccionar espacios libres del tablero que se le pasa.
* devolverSemillas(numSemillas:Int, valores:List[Int]):List[Int] =: Devuelve la lista de las distintas semillas que se van a generar.
* ponerSemillas(tablero: List[Int], semillas: List[Int]): List[Int] =: Función encargada de introducir las semillas de la lista de semillas en el tablero.
* poner(num:Int, pos:Int, tablero:List[Int]): List[Int]=: Función que añade el número indicado en la posición especificada del tablero. Devuelve el tablero resultante.
* imprimir\_cabeceras(dificultad:Int)= dificultad match: Imprime las cabeceras del tablero.
* imprimir\_tablero(filas:Int, fil:Int, columnas:Int, col:Int, dificultad:Int, tablero:List[Int]): Unit=: Imprime el tablero, con los delimitadores correspondientes.
* moverGen(tablero: List[Int], columnas: Int, posicion: Int):List[Int] =: Función general para mover hacia la derecha y sumar. El resultado lo vuelve a desplazar por si han quedado huecos.
* moverDer (tablero: List[Int], columnas:Int, posicion:Int):List[Int] =: Función que selecciona cada fila por separado y llama a moverDerAux para realizar el movimiento.
* moverDerAux(tablero: List[Int], columnas: Int, posicion: Int):List[Int] = Desplaza la lista hacia la derecha.
* sumar(tablero: List[Int], columnas: Int, posicion: Int): List[Int]=: Realiza la suma de las celdas colindantes (en función del movimiento).
* moverIzq(tablero: List[Int], columnas: Int, posicion: Int):List[Int] =: Mueve a la derecha los números de la lista invertida, e invierte el resultado. Todo esto para que el resultado sea el mismo que haber movido hacia la izquierda.
* moverAbajo(tablero: List[Int], columnas: Int, tam: Int, posicion: Int):List[Int] =: Realiza la matriz traspuesta del tablero, mueve el resultado a la derecha y hace la traspuesta del resultado. Todo esto para que el resultado sea el mismo que haber movido hacia abajo.
* moverArriba(tablero: List[Int], columnas: Int, tam: Int, posicion: Int):List[Int] =: Realiza la matriz traspuesta del tablero, mueve el resultado a la izquierda y hace la traspuesta del resultado. Todo esto para que el resultado sea el mismo que haber movido hacia arriba.
* traspuesta (tablero:List[Int], columnas: Int, tam:Int, pos:Int):List[Int]=: Función que hace la traspuesta de la matriz introducida.
* cogerN(n:Int, l:List[Int]):List[Int] =: Devuelve la lista de los N primeros elementos de la lista introducida.
* quitar(n:Int, l:List[Int]):List[Int] =: Función que quita los N primeros elementos de la lista introducida.
* coger(n:Int, tablero:List[Int]):Int=: Función encargada de coger el elemento que le indiques.
* mover (movimiento:Int, tablero:List[Int], columnas:Int, dificultad:Int, puntuacion:Int) = movimiento match: Función que, en función del movimiento que haya solicitado el jugador, llama a la función correspondiente. Si es 0, termina el juego, y si es otro número que no sea el de los movimientos o el 0, devuelve el mismo tablero para que se intente otra vez.
* tableroLleno(tablero:List[Int]):Boolean =: Función encargada de comprobar si el tablero está lleno. Si es así devuelve true, en caso contrario, false.
* jugar(tablero:List[Int], filas:Int, columnas:Int, dificultad:Int, malo: Boolean, semillas: Int): Unit =: Función principal de juego. Imprime el tablero de juego, pide los movimientos, comprueba si estos son correctos, llama a los movimientos y genera las semillas para el siguiente turno.