

## INFORME DE PROYECTO II

Para realizar el proyecto utilicé solo un objeto llamado "jugador", el cual tiene por atributos dos enteros (para hacer papel de la coordenada X y la coordenada Y de la matriz) y sus respectivas funciones sets y gets.

Como lo indica el proyecto, al iniciar el programa se piden las dimensiones de la matriz y la cantidad de tiempo que toma una captura a distancia. Se crean:

- 2 arreglos de chares para guardar las soluciones (las soluciones posibles y la solución definitiva),

- 2 matrices, una para simular la ciudad y otra para validar las posiciones por donde ya pasó el jugador.

Se lee por pantalla los caracteres que se insertaran en la matriz, al momento de leerse el carácter 'C' es guardado en la matriz y asignadas las coordenadas iniciales del objeto jugador y al leerse un número es aumentado un contador para saber cuántos pokemons hay en la matriz.

Luego, es llamada la función principal "jugar" donde se empieza a ejecutar el backtracking:

- 1- Se pregunta si desde donde está posicionado el jugador se puede hacer una captura a distancia llamando a la función "capd" que recorre la matriz en dirección vertical y horizontal buscando pokemons y de haber uno devuelve las coordenadas del pokemon para luego capturarlo, guardar en la solución temporal la captura, marcando la casilla como vacía y llamando al backtracking nuevamente.
- 2- Se pregunta si donde está posicionado el jugador hay un pokemon, de haber uno lo captura, guarda en la solución temporal, marca la casilla como vacía y llama nuevamente al backtracking.
- 3- Se pregunta si es válido el movimiento del jugador hacia arriba, abajo, derecha o izquierda, sumándole 1 al tiempo por cada movimiento, marcando la matriz auxiliar de que pasó por esa posición y guardando la dirección en la solución temporal y llamando de nuevo a la función para el backtracking.

- 4- En el momento de que no se puedan hacer ninguno de los movimientos anteriores mencionados, se envía el arreglo de solución temporal, junto a los tiempos y el arreglo de solución final a la función “guardar” para validar cual es una mejor solución.

Todo este proceso se va a repetir mientras queden pokemones en el tablero y mientras la solución temporal sea menor a la mejor solución que llevamos en ese momento, ya que si la solución temporal sobrepasa a la solución óptima que llevamos, indica que es un camino más largo por lo cual no debería continuar.

Cuando ya se haya obtenido el mejor recorrido con menor tiempo, es impreso por pantalla dicha solución.