

Proyecto 2: Pokémon Go en Venezuela

Recientemente ha salido la aplicación de Pokémon Go en Venezuela y ya está disponible para su descarga de manera gratuita en dispositivos móviles. Pokémon Go es un juego de realidad aumentada que consiste en recorrer el mundo e ir atrapando criaturas llamadas Pokémon, para esto los jugadores deben salir a las calles en su búsqueda. Sin embargo debido a ciertos inconvenientes que ocurren en el país, es bastante complicado salir a la calle y jugar Pokémon Go con tranquilidad. Es por esta razón que muchos jugadores se han planteado la creación de una aplicación que les ayude a capturar el mayor número de criaturas en el menor tiempo posible, para así evitar (lo más que se pueda) cualquier problema al salir a jugar Pokémon Go, sin dejar de disfrutar de la misma.

Parte de la aplicación ya está realizada, sin embargo se quiere que usted desarrolle el algoritmo que indique la forma en que un jugador pueda atrapar a los Pokémon en una ciudad correspondiente, en el menor tiempo posible. La aplicación representa la ciudad como una matriz de tamaño $N \times M$, representando las calles como filas (N) y las avenidas como las columnas (M). Cada cruce entre una calle y una avenida (es decir una posición de la matriz) representa un punto de interés que puede ser lo siguiente:

- El carácter “.” (el punto), representa que el cruce está vacío y el jugador puede transitar en él.
- El carácter “*” (el asterisco), representa un edificio y por lo tanto el jugador no puede atravesarlo.
- Un entero 0-9 que indica la presencia de un Pokémon con un rango de distancia desde el cual es posible atraparlo.
- El carácter “C” (letra C) que indica la posición donde inicia el jugador.

El jugador puede moverse solo a través de las calles y avenidas adyacentes, es decir solamente una posición en dirección horizontal y vertical, este movimiento siempre le tomara 1 minuto de su tiempo.

Para poder atrapar un Pokémon es necesario estar en la misma posición en donde se encuentra el Pokémon, este tipo de captura no requiere costo alguno puesto que al estar tan cerca, es bastante sencillo capturar al Pokémon. Sin embargo, algunos se pueden atrapar a distancia (según la que indique el Pokémon), pero el jugador debe estar posicionado en la misma calle o avenida que el Pokémon, es decir debe poder verse directamente tanto en horizontal como en vertical y en el medio del jugador y el Pokémon no puede estar un edificio u otro Pokémon. Por supuesto capturar a distancia requiere un costo adicional, sin embargo se considera que el jugador tiene un límite de T minutos que puede capturar el Pokémon, por lo que cada vez se capture un Pokémon a distancia se deberá sumar T minutos al tiempo total que le tome capturar todos los Pokémon. Cabe destacar que si el rango de distancia de un Pokémon es 0, este solo será posible capturarlo en la misma posición donde se encuentra.

Entrada

La primera línea consta de 3 enteros N, M y T indicando el número de calles y avenidas que tiene la ciudad, además del tiempo que cuesta atrapar un Pokémon a distancia. A continuación vendrán N líneas con M caracteres las cuales representan a la ciudad según lo descrito anteriormente.

Salida

En la primera línea se debe imprimir el tiempo mínimo total que le costó al jugador capturar todos los Pokémon. En la segunda línea se deben describir los pasos que corresponde a la forma mínima de capturar a los Pokémon, para los movimientos se deben usar los caracteres “N” (movimiento hacia el norte), “S” (al sur), “E” (este), “O” (oeste), para indicar la captura a distancia se deberá imprimir el carácter “C” seguido de la dirección en la que se realizó la captura (Por ejemplo si se captura el primer Pokémon al norte se imprimiría “CN”), si la captura se hace en la misma posición donde está el pokémon se imprime solamente el carácter “C”. Todos estos movimientos deben estar separados por un espacio. Si existen varias soluciones que tomen el mínimo tiempo,

entonces deberá imprimir la menor lexicográficamente de la cadena que representan los pasos de la salida.

Ejemplos

Entrada	Salida
4 4 1 .3.. .2..C..	2 CN CN
4 4 1 .30.. .2..C..	4 N N C N C E C
5 4 2 C**2 .**. .**. *****	9 S S S E E E N CN

Para el primer, cuesta menos capturar a los Pokémon desde lejos, sin embargo en el segundo ejemplo conviene atraparlos a todos acercándose puesto que igual se tiene que subir a capturar al que tiene rango de captura 0.

Para el tercer ejemplo hay 2 soluciones posibles se puede intentar capturar al Pokémon a distancia (desde la posición 2,3 de la matriz, indexada desde 0) o se puede ir directamente por el Pokémon, sin embargo la cadena correspondiente a capturarlo a distancia (S S S E E

E N CN) es menor lexicográficamente que la cadena para irlo a capturar directamente (S S S E E E N N N C), por tanto se escoge la primera solución.

Consideraciones:

- La solución al proyecto debe ser realizada utilizando la técnica de Backtraking.
- La matriz de entrada solo contendrá los caracteres descritos, sin ningún espacio o algún carácter no especificado.
- Una vez que el jugador capture un Pokémon, el cruce donde este estaba ubicado se considera como vacía, al igual que el cruce donde comienza el jugador.
- La entrada siempre será válida, es decir, siempre se podrá capturar todos los Pokémon y solo habrá exactamente un carácter “C” indicando la posición de inicio del jugador.
- La entrada y salida debe ser estricta, es decir su programa debe leer los datos tal y como se especifican, e imprimir las respuestas tal como se especifica en el ejemplo. Cualquier información adicional que solicite o proporcione será penalizada.
- La lectura y escritura de datos debe ser hecho usando la Entrada/Salida estándar de C++.
- El programa debe ser realizado en C++ sin el uso de STL para las estructuras de datos, a excepción de la clase **string** provista por C++.
- Debe emplear Programación Orientada a Objetos en su programa.
- El código fuente debe estar intradocumentado adecuadamente. Se debe entregar un informe en formato PDF con la explicación de la solución de su proyecto (sin incluir código), explicación de las estructuras usadas, la complejidad de cada una de las funciones descritas y la identificación del autor. La ausencia de este archivo califica al proyecto como no entregado.
- Será compilado y evaluado en el compilador g++ (Linux). Si el proyecto no compila no será corregido.
- El proyecto debe ser realizado de manera individual.
- La revisión del proyecto no es obligatoria y solo se revisaran proyectos que el grupo docente considere necesario.

**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
ALGORITMOS Y ESTRUCTURAS DE DATOS**

- El proyecto debe entregarse el día 28 de octubre de 2016 a través de la página de la asignatura, hasta las 11:59 pm como hora límite. No se aceptarán proyectos fuera de las horas establecidas para ello, y se considerarán como no entregados.

Las copias entre proyectos tendrán una calificación de 0 puntos, además de una sanción para todos los involucrados