

Estructuras de Datos.
Facultad de Ciencias, UNAM 2015-2.
Práctica 02: Buscaminas.

Armando Ballinas Nangüelú.
armballinas@gmail.com

Fecha de entrega jueves 19 de febrero de 2015.

1. Introducción

El objetivo de esta práctica es que el alumno desarrolle habilidades para resolver problemas y se familiarice con la estructura y el tipo de problemas en concursos de computación.

El problema a resolver es muy simple: dada la representación de un tablero del juego "Buscaminas", en inglés *MineSweeper*, decir cuántas minas adyacentes tiene cada casilla.

2. Buscaminas

El juego Buscaminas formaba parte de la colección de juegos por defecto en los sistemas windows a finales del siglo pasado y a principios de este. El juego consistía de un tablero de casillas (celdas), en donde el jugador debía dar click.^a alguna de ella. La celda podía tener una mina o no, en el primer caso el juego concluía y el jugador había perdido, en el segundo, la casilla descubierta mostraba un número. Este número correspondía a la cantidad de minas adyacentes a esta celda. Usando la información de las celdas descubiertas el jugador debería descubrir todas las celdas que no contengan una mina en el tablero para poder ganar.

3. El problema a resolver

El problema consiste en decir cuántas minas adyacentes tiene cada casilla en un tablero del juego.

3.1. El tablero

El tablero será una serie de renglones. Cada renglón tendrá un caracter por cada celda del tablero en ese renglón. Si en la celda hay una mina habrá el caracter asterisco (*), de lo contrario estará el caracter punto (.). Entonces un tablero de 4 x 4 con 5 minas se puede representar de la siguiente manera:

```

. . * .
. * . .
. . * *
. . * .

```

3.2. La entrada

La entrada del programa debe ser desde la entrada estándar. La entrada estará compuesta por varios casos de prueba.

Cada caso de prueba es el tamaño de un tablero de juego seguido de la representación del tablero. Y se compone de la siguiente manera:

- La primera línea serán dos números n y m separados por un espacio.
- Después habrá n líneas, en donde cada línea contendrá m caracteres. Cada uno de estos caracteres podrán ser . o *.

El fin de la entrada estará dado por el fin de archivo o por el tablero vacío, es decir, el tablero con cero renglones y cero columnas.

Su programa debe lanzar mensajes de error apropiados cuando la entrada se presente en un formato erróneo. Algunos casos para lanzar mensajes de error son:

- Hay un caso de prueba que esté incompleto, es decir, que tenga más renglones declarados que los leídos.
- Un tablero tiene caracteres no válidos. Es decir, tiene caracteres distintos a "*" o ".".
- El fin de archivo se encuentra inesperadamente mientras se lee un tablero.

3.3. Salida

La salida debe ser, por cada caso de prueba:

- La cadena "Field #N:" Donde "N" es el número de caso de prueba.
- Después se debe imprimir el tablero de entrada, sustituyendo cada casilla sin mina por el número de minas adyacentes a ella. **Solo** se deben sustituir las casillas sin mina. Las casillas con mina deben dejarse igual, es decir, mostrarse con el caracter '*'.
- Después de cada caso de prueba se debe dejar una línea en blanco.

se debe imprimir un número por celda del tablero. Este número representará las minas que están adyacentes a esa celda. Por ejemplo para entrada siguiente con dos casos de prueba:

```

4 4
*...
....
.*..
....
3 5
**...

```

```
.....
.*...
0 0
```

Se debe producir la siguiente salida:

Field #1:

```
*100
2210
1*10
1110
```

Field #2:

```
**100
33200
1*100
```

4. Evaluación

El objetivo de la práctica no solo es resolver el problema para cada caso de prueba, sino que el alumno se familiarice con las restricciones que se presentan en la entrada y salida.

Si el formato de entrada o salida no es el precisado en este documento se penalizará al alumno.

Por otro lado, si la solución de la práctica es muy buena (a consideración del ayudante) se obtendrán puntos extra.

5. Ejercicios

1. (1 pt.) Entregar de forma escrita el día de la entrega de la práctica al inicio de la clase de laboratorio la respuesta a las siguientes preguntas:
 - a) Describir el algoritmo usado para resolver el problema, diciendo qué es lo que se hace en cada paso del algoritmo y cuántas veces se repiten.
 - b) Dar la complejidad de su algoritmo en notación asintótica ($O(f(n, m, p))$) donde f es una función en términos del tamaño del tablero (n, m) y del número de minas (p) , es decir la complejidad dada debe depender de la dimensión del tablero y del número de minas que contiene. Así mismo deben decir porqué su algoritmo efectivamente cumple con esta cota.
2. (8 pts.) Resolver el problema del juego de buscaminas descrito en este documento respetando los formatos para entrada y salida.
3. (1 pt.) Documentar su solución utilizando los comentarios de Java. No es necesario utilizar javadoc, pero su código sí debe estar documentado.

6. Formato de entrega

Se deben entregar al menos los siguientes archivos:

1. `Main.java`. En este archivo debe ir todo el código necesario para resolver el problema. Solo se debe entregar este archivo con código.
2. `Readme.txt`. Un archivo Readme en donde deben escribir su nombre, número de cuenta y correo así como la manera de compilar y ejecutar su programa.

Las prácticas se deben enviar al correo **armballinas@gmail.com** con el asunto: **[eddp2]**.