

TP1
IFT2015
Structures de données

Bomb Exploder

On vous donne une grille initiale de $n \times m$ bombes, dont le statut est indiqué par une valeur entière $\in [0..2]$. La valeur 2 indique une bombe active ; 1 bombe inactive ; 0 pas de bombe. La grille initiale est donnée dans un fichier ascii, où vous trouverez en première ligne les valeurs n et m , puis l'état des bombes, une matrice $n \times m$. Voici un exemple:

```
3 3
0 1 2
1 2 0
2 1 1
```

À chaque itération, les bombes actives explosent (indiquées par les 2 soulignés) et activent toute bombe inactive adjacente (les 1 qui deviennent des 2). Les positions adjacentes d'une position sont celles situées à sa gauche, à sa droite, en haut ou en bas. Après la première itération, les bombes aux positions (0,2), (1,1) et (2,0) ont explosé et les bombes inactives adjacentes sont devenues actives. La grille ressemble maintenant à ceci :

```
0 2 2
2 2 0
2 2 1
```

On continue tant que les bombes inactives peuvent être activées. Ici, après la deuxième itération, la grille devient :

```
0 2 2
2 2 0
2 2 2
```

Et, enfin, après la troisième itération ceci :

```
0 2 2
2 2 0
2 2 2
```

Votre programme doit retourner le nombre d'itérations nécessaires pour faire exploser toutes les bombes, c'est-à-dire dans cet exemple **3**.

S'il reste des bombes inactives dans la grille, votre programme doit retourner la valeur **-1**. Voici un tel exemple :

```
2 3
1 2 0
0 0 1

2 2 0
0 0 1
```

```

2 2 0
0 0 1

```

La bombe inactive dans le coin inférieur droit est protégée par des positions adjacentes inoccupées, et donc cette bombe ne devient jamais active.

Voici un autre exemple qui se résolve en 4 itérations :

```

5 3
2 2 0
1 1 0
0 0 0
2 1 1
1 1 1

```

```

2 2 0
2 2 0
0 0 0
2 2 1
2 1 1

```

```

2 2 0
2 2 0
0 0 0
2 2 2
2 2 1

```

```

2 2 0
2 2 0
0 0 0
2 2 2
2 2 2

```

```

2 2 0
2 2 0
0 0 0
2 2 2
2 2 2

```

La résolution de cette grille a nécessité 4 itérations, donc votre programme retourne la valeur **4**.

Fichier d'entrée

Il n'y aura pas d'erreur dans le fichier d'entrée, qui sera transmis à votre programme comme premier argument (c'est-à-dire `args[0]` dans les arguments de la fonction `main`). De nombreuses grilles peuvent être décrites dans un seul fichier d'entrée. Le fichier d'entrée `sample.txt` fourni avec le TP contient les trois exemples ci-dessus.

Soumission et code Java

- Vous pouvez former des équipes de 2 programmeurs ou moins.

- Pour résoudre ce problème, vous devez suivre les règles de la POO ; utilisez `CamelCase` pour vos identificateurs ; et commentez votre code.
- Vous devez utiliser une file d'attente, que vous allez soit implémenter vous-même, soit prendre directement dans les codes du cours. Les méthodes que vous utiliserez sont : `enqueue(x)`, `dequeue()`, `size()` et `isEmpty()`. Aucune autre méthode n'est autorisée !
- Lors de votre remise, vous devez mettre tous les fichiers `Java` et les fichiers d'entrée doivent dans le même répertoire que vous nommerez TP1. Ne pas utiliser la clause `package`.
- Vous devez nommer la classe qui contient la fonction principale (main) `BombExploder`.
- Vous devez archiver votre répertoire (seuls les formats `TP1.tgz` et `TP1.zip` seront autorisés).
- Vous avez jusqu'au 28 octobre 23h59 pour soumettre votre solution sur StudiUM.
- Si vous ne remettez pas votre solution à temps, vous recevrez une pénalité de 20% par période de 24h commençant à 00h00 chaque jour à partir du 29 octobre.
- Vous devez suivre le format de sortie exact, c'est-à-dire un entier par ligne (un pour chaque problème dans le fichier d'entrée sans ligne blanche ni avant ni après les résultats).

Évaluation

- 10% si votre code compile correctement sans aucun argument (`javac BombExploder.java`)
 - 10% si votre code exécute et trouve les résultats corrects sur les exemples fournis (voir l'Annexe ci-dessous).
 - 40% si votre code exécute et trouve les résultats corrects sur des exemples jamais vus et déterminés par le correcteur.
 - 20% si votre code est efficace (comment cela sera fait reste à déterminer)
 - 20% si votre code suit les principes de la POO, est lisible et contient des commentaires appropriés.
- ★ Vous devez avoir 50% ou plus dans les examens pour que les points des devoirs comptent.
- ★ N'oubliez pas d'inclure vos noms et numéros de matricule dans toutes vos classes Java.
- ★ Si vous utilisez le code du cours, copiez-le dans votre répertoire et supprimez la déclaration de package en haut.

Questions

Si vous avez des questions, utilisez le forum TP1 sur StudiUM, ou contactez un des démonstrateurs ou le Prof.

Amusez-vous !!!

Annexe

Étant donné le fichier d'entrée fourni (`sample.txt`):

```
2 3
1 2 0
0 0 1
3 3
2 2 1
1 1 0
1 1 2
5 3
2 2 0
1 1 0
0 0 0
2 1 1
1 1 1
```

l'appel de votre programme : `java BombExploder sample.txt`
doit produire la sortie :

```
-1
3
4
```

PS. Pas de ligne blanche avant ou après ces trois lignes.