

La viande des gris zombies

1 le background

Vingt ans après le volume 4 réalisé par le dieu du calembour foireux, les films de zombies reviennent à la mode. Un producteur au nez creux vous retrouve sur « mes copains d'avant » et vous embauche directement comme réalisateur, puisque vous avez si bien oeuvré auparavant. Malheureusement, vous êtes une bille en scénario et la seule idée que vous trouvez est de reprendre le pitch des deux premiers, mais avec comme acteurs des paysans du Larzac rejoignant leur coopérative laitière et des brebis zombies. Vous misez donc tout sur le fait qu'il y aura cette fois plein d'humains, des montagnes de zombie et que de beaux effets spéciaux combleront la vacuité du scénario. En effet, on peut se demander qu'est-ce que les brebis font sur la route alors qu'elles devraient rester bien sagement dans les champs à brouter l'herbe sèche du Causse. Une brebis, c'est déjà très con, mais alors quid d'une brebis zombie ? Quant aux paysans, pourquoi doivent-ils rester sur la route de la coopérative et se cacher derrière les murets en pierre sèche, plutôt que gambader librement sur le Causse totalement désert et ainsi survivre ?

2 l'énoncé

Pour tester différents scénarios, vous embauchez un stagiaire de BUT, bien évidemment sans rémunération, faut pas exagérer. Les différentes situations sont décrites par une chaîne de caractères de longueur N (variable selon les situations, mais $N \leq 10000$) composée de points, de caractères C pour les cachettes, de 2 à N caractères H représentant les humains, et de 0 à $N - 2$ caractères Z représentant les brebis zombies. Les ., C, H et Z peuvent se trouver n'importe où dans cette chaîne mais **jamais** au même endroit. Par exemple : H.H.CZC..Z.H ou encore ZHZH.Z.C. Pour simuler le déplacement des intervenants, on utilise un pas de temps plutôt qu'un déplacement en continu. A chaque pas de temps, on suppose que les Z se déplacent de 1 caractère vers la gauche, les H de 1 caractère vers la droite ou 0 s'ils se cachent. L'objectif des H est d'atteindre la fin de chaîne qui représente l'arrivée, **le plus vite possible**. La simulation doit déterminer si chaque H va :

- croiser un zombie sans être caché, et donc mourir,
- atteindre l'arrivée, soit sans jamais se cacher, soit en se cachant **ET** en sortant des cachettes le plus tôt possible.

Important : les déplacements étant concurrents, on considère que l'humain peut se cacher s'il atteint une cachette exactement en même temps que le zombie. Il peut

également atteindre la cachette avant et s’y cacher pendant plusieurs tours, c’est-à-dire le temps que le zombie atteigne lui-même la cachette. En revanche, si le zombie est juste après l’humain et que ce dernier bouge, alors il meurt forcément. Ces situations sont illustrées par les exemples suivants.

- H.C.Z : après 2 déplacements, H et Z arrivent au même moment sur C. H peut donc se cacher pendant tout un tour et survivre.
- HC.Z : après 1 déplacement, H arrive sur C et Z juste après. Le tour suivant, si H sort de la cachette, il rencontre Z et meurt. H doit donc se cacher pendant deux tours pour survivre.
- H.CZ : après 1 déplacement, Z arrive sur C, et H juste avant. Lors du déplacement suivant, H pourrait arriver sur C. Mais comme Z avance aussi, H n’a pas le temps de se cacher et meurt.

Pour tester les situations, votre programme doit lire sur l’entrée standard :

1. un entier M représentant le nombre de situations à tester
2. M chaînes de caractères au format décrit ci-dessus (avec ., C, H, Z), représentant chacune une situation.

Ensuite, votre programme doit déterminer pour chaque situation, le sort de l’humain et écrire sur la sortie standard :

- une suite de valeurs entières $X \geq -1$. La i^{eme} valeur représente le sort du i^{eme} humain dans la chaîne, en partant de la gauche. S’il meurt X vaut `dead`. S’il peut atteindre l’arrivée, X est le nombre de pas de temps pour atteindre le dernier caractère de la chaîne (donc 0 s’il y est dès le début).

Comme dans les problèmes précédents, il faut déterminer les valeurs X en considérant que si un humain est caché dans une cachette, il doit en sortir dès qu’il lui est possible d’atteindre une autre cachette sans être tué. Attention, il est possible que plusieurs humains puissent se trouver en même temps dans la même cachette.

Le tableau 1 donne un exemple d’entrée et la sortie associée.

entrée	sortie
4	14 4 0
.H.C..ZZ.H..CH	dead dead 11
H...H..H.C.Z..Z.	28 28 14
.HH.C..Z.HCC.....Z	dead 16 16 14 14 12
HHHHHCCCCZZZ	

TABLE 1 – Exemple d’entrée et de sortie associée

Commentaires :

- situation 1 : l’humain 1 met 2 pas de temps pour arriver dans la cachette 1. Il attend ensuite 2 pas de temps que le zombie 2 le rejoigne. Il lui faut ensuite 10 pas de temps pour rejoindre l’arrivée soit 14 au total. L’humain 2 met simplement 4 pas de temps pour arriver et l’humain 3 est déjà à l’arrivée donc son temps = 0.

- situation 2 : les deux premiers humains n'auront jamais le temps d'arriver à la cachette : ils meurent. L'humain 3 met 2 pas de temps pour arriver à la cachette, puis attend 3 pas que le deuxième zombie le dépasse. Il lui faut ensuite 6 pas de temps pour atteindre l'arrivée. Au total, il met 11 pas de temps.
- situation 3 : les deux premiers humains se rejoignent au bout de 3 pas de temps dans la cachette 1, au même moment que le zombie 1. Ils n'ont pas le temps d'arriver à la cachette 2 avant l'arrivée du zombie 2 et doivent rester cachés 11 pas de temps. Ensuite, ils mettent 14 pas pour atteindre l'arrivée, soit un total de 28.
- situation 4 : à vous de faire l'interprétation :-)

IMPORTANT : le temps limite pour calculer les solutions des M situations est de 20 secondes. C'est normalement suffisant pour trouver les solutions du plus gros fichier secret si votre programme est un minimum optimisé.

3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur

<http://cours-info.iut-bm.univ-fcomte.fr>

un article dans la section **hackathon** portant le même titre que l'exercice. Il contient un lien permettant de télécharger un canevas de code, ainsi que le fichier d'entrée donné ci-dessus.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.