



Portal

Titus Cieslewski

Medium Task

Well, that was not the exit from cyberspace, but just a gate to another such room. But this time, some things have changed: Firstly, the wormhole is now on top of a cube, and secondly, there are now plenty of other cubes, all in the same plane as the portal behind Heidi and the gate cube. Now that makes things trickier...

Same as before. The position of obstacle cubes will be given with integer coordinates. When Heidi hits a cube side, she will fall down straight and hit the next lower cube whose x coordinate is one to the left of the one of the cube that she hit. Given entering speeds, return which of them will make Heidi land on or hit a cube such that she falls into the wormhole or is able to walk into the wormhole. Heidi is able to walk from a cube c_i to a cube c_j if there exists a $dx \in \{-1, 1\}$, such that $x_{c_i} + dx = x_{c_j}$ and $y_{c_j} \geq y_{c_i}$ and there exists no other cube c_k with $x_{c_k} = x_{c_j}$ and $y_{c_i} - 1 \leq y_{c_k} < y_{c_j}$.

Write a function `int check_speed(double s)`, that returns a nonzero value if the supplied speed s would help Heidi reach the wormhole. Read out the problem data in the function `void initialize(cube g, int N, cube o)`, where cube g hosts the wormhole on its top, and there are N cubic obstacles o .

There will be at most 1'000 obstacles and 1'000 speed queries, and the cube coordinates will be between 1 and 500.

Tâche Moyenne

Et bien, on dirait bien qu'il ne s'agissait pas de la sortie du cyberspace, mais juste une porte vers une autre pièce du même genre. Cette fois, cependant, certaines choses ont changé: tout d'abord, le trou de ver est maintenant sur la face supérieure du cube et, deuxièmement, il y'a maintenant un grand nombre d'autres cubes, tous sur le même plan que le portail derrière Heidi et le cube-trou de ver. Voilà qui complique bien les choses...

La position des cubes-obstacles vous est donnée en coordonnées entières. Lorsque Heidi est projetée contre le coté d'un cube, elle va tomber à la verticale et atterrir sur un cube plus bas, dont la coordonnée x est inférieure d'une unité à celle du cube contre lequel elle a été projetée. Étant données les vitesses de départ, retournez laquelle permet à Heidi d'atterrir sur ou d'être projetée contre un cube de telle sorte à ce qu'elle tombe dans le trou de ver où qu'elle soit à même de marcher jusqu'à ce dernier. Heidi peut marcher d'un cube c_i à un cube c_j s'il existe un $dx \in \{-1, 1\}$, tel que $x_{c_i} + dx = x_{c_j}$ et $y_{c_j} \geq y_{c_i}$ et s'il n'existe aucun autre cube c_k avec $x_{c_k} = x_{c_j}$ et $y_{c_i} - 1 \leq y_{c_k} < y_{c_j}$.

Ecrivez une fonction `int check_speed(double s)`, qui retourne une valeur non-nulle si la vitesse fournie s permet à Heidi d'atteindre un trou de ver. Lisez les données du problèmes dans la fonction `void initialize(cube g, int N, cube o)`, où la face supérieure du cube g est couverte par le trou de ver et où il y a N obstacles cubiques o .

Il y aura au plus 1'000 obstacles et 1'000 demandes relatives à différentes vitesses, et les coordonnées des cubes seront comprises entre 1 et 500.

Mittlere Aufgabe

Nun gut, das war jetzt nicht der Ausgang aus dem Cyberspace, sondern bloss ein Wurmloch zu einem weiteren solchen Raum. Allerdings haben sich einige Sachen verändert: Erstens, das Wurmloch ist nun auf einem Würfel, d.h. nur von oben zugänglich und zweitens gibt es nun weitere Würfel welche nicht mit einem Wurmloch ausgestattet sind sondern bloss Hindernisse darstellen...

Der Raum ist gleich beschrieben wie vorhin. Die Hindernis- Würfel sind ebenfalls parallel zu den Achsen, haben Einheitsgrösse und sind mit der linken oberen Ecke an ganzzahligen Koordinaten verankert. Wenn Heidi seitlich auf einen Würfel prallt, fällt sie auf den nächstniedrigen Würfel dessen x -Koordinate um 1 kleiner ist als die des Würfels, auf den sie geprallt ist. Des Weiteren kann Heidi auf den Hindernissen laufen, d.h. sie muss nicht zwingend direkt ins Wurmloch fallen, aber es muss einen Pfad dahin geben. Heidi kann von einem Würfel c_i zu einem Würfel c_j gelangen, falls ein $dx \in \{-1, 1\}$ existiert, so dass $x_{c_i} + dx = x_{c_j}$ und $y_{c_j} \geq y_{c_i}$ und es keinen Würfel c_k mit $x_{c_k} = x_{c_j}$ und $y_{c_i} - 1 \leq y_{c_k} < y_{c_j}$ gibt.

Schreibe die Funktion `int check_speed(double s)`, welche 0 ausgibt falls die gegebene Geschwindigkeit Heidi nicht zum Wurmloch bringt. Falls doch, gib eine beliebige Zahl zurück. Die verbleibenden Problemparameter können mit der Funktion `void initialize(cube g, int N, cube o)` abgerufen werden, wobei Würfel g das Wurmloch enthält, und es N Hinderniswürfel o gibt.

Es wird höchstens 1'000 Hinderniswürfel und 1'000 Geschwindigkeitsabfragen geben, und beide Würfelkoordinaten werden jeweils zwischen 1 und 500 liegen.

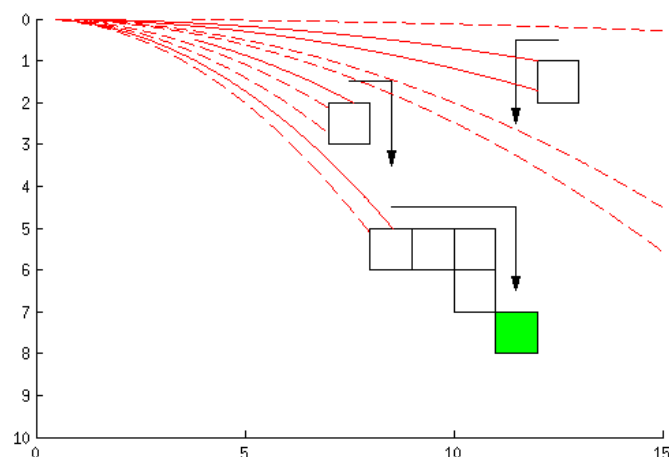


Figure 1:

E: Illustration of the sample test case given in the problem folder. Here, Heidi can walk to the wormhole from all the cubes. The walking paths are indicated with arrows. Full lines represent speeds that allow Heidi to reach the wormhole, dashed lines represent speeds that make it impossible to reach the wormhole.

F: Illustration du cas de test donné dans le dossier du problème. Ici, Heidi peut marcher jusqu'au trou de ver à partir de tous les autres cubes. Les chemins de marche sont indiqués avec des flèches. Des lignes complètes représentent les vitesses qui permettent à Heidi d'atteindre le trou de ver, les lignes pointillées représentent les vitesses qui ne permettent pas de l'atteindre.

D: Darstellung des im Problemordner gegebenen Problems. In diesem Fall kann Heidi von allen Würfeln zum Wurmloch laufen. Die Pfade sind mit Pfeilen angegeben. Volle Linien zeigen Geschwindigkeiten, mit denen Heidi zum Wurmloch gelangen kann, gestrichelte Linien sind schlechte Geschwindigkeiten.