

# Quarantine

Christian Kauth

It was a perfect storm of unfortunate coincidences: Heidi's anti-virus suffered a temporary failure, during which she ate a poisonous leaf, which broadened her pupils. Dazzled then by a laser beam, she rushed out of a portal with illogical speed and crashed against an unallocated memory block – ouch! When waking up again, she found herself in the quarantine division of cyberspace, where access to almost everything fun is restricted.

Heidi begs you to help her getting out of that boring place as soon as possible. To do so, she has to prove her sanity by passing a vision test. That's not as easy as it sounds, as Heidi suffers serious hallucinations since her accident and sees virtual bits all over the place! Consider Heidi's vision field as a regular 2D matrix of  $N$  by  $N$  pixels, each either shining in flashy colours (logical 1) or remaining dark (logical 0). To pass the vision test, Heidi will have to answer correctly some queries about the square  $N \times N$  pictures she is presented,  $1 \leq N \leq 1000$ .

## Easy Task

Before being able to answer any question, write a function

`void load_picture(bool p[1000][1000], int N)` to copy the hallucinated picture from Heidi's brain.  $p$  is the picture that contains 0 and 1 values and both dimensions span 0 to  $N - 1$ .  $p[r][c]$  is the status of the pixel in row  $r$  and column  $c$ . The doctor of the quarantine section will then start his queries by calling the function `bool query(int r, int c)`. This function must return `true` if the majority of the pixels that lie either in row  $r$  or column  $c$  or both, are 1, and `false` otherwise.  $r$  and  $c$  are guaranteed to lie within the picture. There will be 10 pictures (calls to `load_picture`) and a cumulative total of  $10^7$  queries to be answered within seconds.

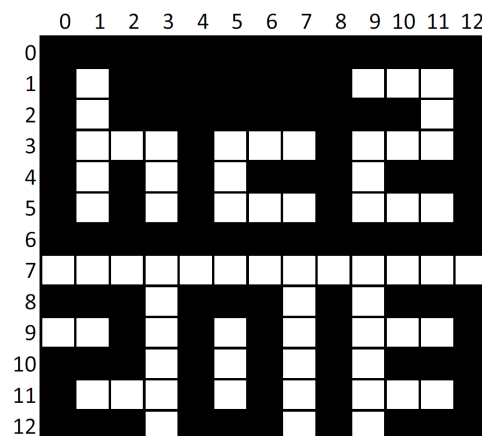


Figure 1: A sample  $13 \times 13$  vision grid. White pixels are 1, black ones are 0.

For the example of Figure 1, `query(1,1)` shall return `false` and `query(7,6)` shall return `true`.

# Quarantine

Christian Kauth

Ce fut une parfaite combinaison de regrettables coïncidences: l'antivirus de Heidi a souffert d'une panne temporaire, durant laquelle elle a mangé une feuille empoisonnée qui lui a dilaté les pupilles. Surprise ensuite et aveuglée par un faisceau laser, elle a couru avec une vitesse illogique droit dans un portail qui l'a projetée contre un bloque de mémoire non-allouée – aïe ! Après avoir repris ses esprits, Heidi se retrouve dans la zone de quarantaine du cyberspace, où l'accès à presque toutes les ressources amusantes est interdit.

Heidi vous supplie de l'aider à sortir de cet endroit barbant aussitôt que possible. Pour ce faire, elle doit prouver sa santé mentale en passant un test de vision. Malheureusement, ça n'est pas aussi facile que ça en a l'air, car Heidi souffre de grosses hallucinations depuis son accident et voit des bits virtuels partout dans son champ de vision. Considérez son champ comme une matrice régulière 2D de  $N$  par  $N$  pixels, chacun étant soit sombre (valeur logique 0) soit brillant avec des couleurs flashy (valeur logique 1). Pour passer le teste de vision, Heidi doit répondre correctement à certaines questions concernant les images carrés  $N \times N$  qui lui seront présentées,  $1 \leq N \leq 1000$ .

## Tâche Facile

Avant de pouvoir répondre à quelque question que ce soit, veuillez écrire une fonction `void load_picture(bool p[1000][1000], int N)` qui copie l'image hallucinée à partir du cerveau de Heidi.  $p$  est l'image qui contient des 0 et des 1 et les deux dimensions s'étendent de 0 à  $N - 1$ .  $p[r][c]$  est le statut d'un pixel sur la ligne  $r$  et dans la colonne  $c$ . Le docteur de la zone de quarantaine va commencer ses demandes avec la fonction `bool query(int r, int c)`.

Cette fonction doit retourner `true` si la majorité des pixels qui sont à la ligne  $r$ , ou dans la colonne  $c$  ou dans les deux, sont 1, et `false` autrement. Il est garanti que  $r$  et  $c$  sont limités au domaine de l'image. Il y aura 10 images (appels à `load_picture`) et un total (cumulé) de  $10^7$  questions auxquelles il faudra répondre en quelques secondes.

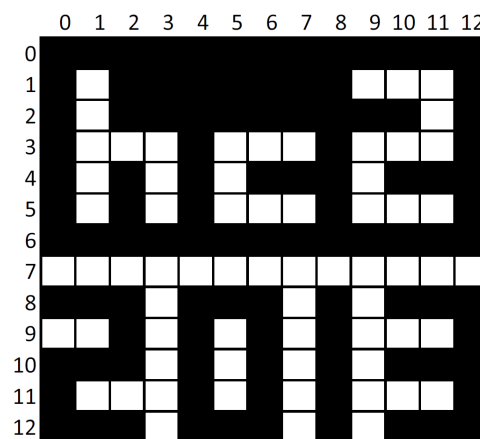


Figure 1: Un exemple de vision (format  $13 \times 13$ ). Les pixels blancs sont 1, les noirs sont 0.

Pour l'exemple de la Figure 1, `query(1,1)` doit retourner `false` et `query(7,6)` doit retourner `true`.

# Quarantine

Christian Kauth

So ein Pech aber auch! Durch eine Kette von unglücklichen Zufällen fiel Heidis Antivirus aus. Sie ass ein giftiges Blatt von einem Baum, und ihre Pupillen weiteten sich. Gerade in diesem Moment wurde sie von einem Laserstrahl geblendet, fiel verwirrt aus einem Portal, und krachte mit unangemessener Geschwindigkeit gegen einen unzugewiesenen Speicherblock – uff! Als Heidi wieder zu sich kam, befand sie sich in der Quarantäneabteilung des Cyberspace. . . ein ziemlich langweiliger Ort.

Bitte hilf Heidi, so schnell wie möglich da raus zu kommen! Dazu muss Heidi einen Sehtest bestehen. Das ist gar nicht so einfach, denn seit ihrem Unfall leidet Heidi an schweren Halluzinationen: sie sieht überall virtuelle Bits. Heidis Sichtfeld ist eine 2D-Matrix mit  $N$  mal  $N$  Pixel, welche entweder in allen Farben schillern (Bit 1) oder dunkel bleiben (Bit 0). Um den Sehtest zu bestehen, muss Heidi einige Fragen über quadratische  $N \times N$  Bilder beantworten ( $1 \leq N \leq 1000$ ).

## Einfache Aufgabe

Bevor du Heidis Fragen beantwortest, schreibe eine Funktion

`void load_picture(bool p[1000][1000], int N)`, welche das halluzinierte Bild aus Heidis Gehirn lädt.  $p$  ist das Bild. Die Koordinaten der Pixel reichen von 0 bis  $N - 1$ .  $p[r][c]$  ist das Pixel in Zeile  $r$  und Spalte  $c$ , und ist entweder 0 oder 1. Der Doktor der Quarantäneabteilung befragt Heidi mittels der Funktion `bool query(int r, int c)`. Diese Funktion muss `true` zurückgeben, wenn die Mehrheit aller Pixel in Zeile  $r$  und/oder Spalte  $c$  1 sind. Ansonsten gibt die Funktion `false` zurück. Die Koordinaten  $r$  und  $c$  sind garantiert innerhalb des Bildes. Die Tests bestehen aus 10 Bildern (Aufrufen von `load_picture`) und insgesamt  $10^7$  Fragen, welche euer Programm innert Sekunden beantworten muss.

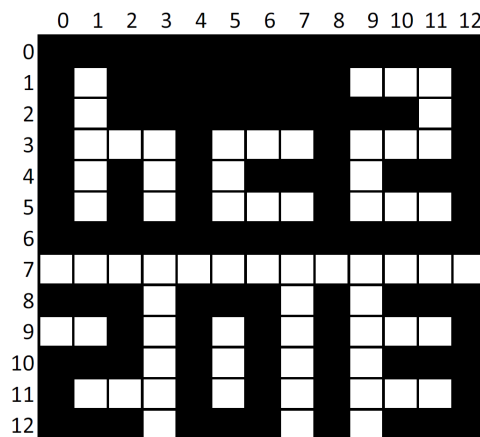


Abbildung 1: Eine Beispielmatrix mit  $13 \times 13$  Pixeln. Weisse Pixel sind 1, schwarze 0.

Für das Beispiel in Abbildung 1 soll `query(1,1)` `false` zurückgeben, und `query(7,6)` soll `true` zurückgeben.