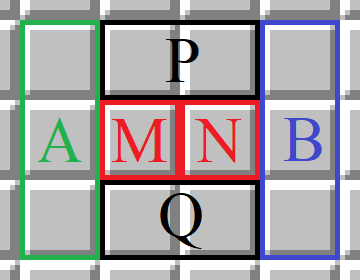
IA

Analyse de base

Vérifier si toutes les cases avoisinantes d’une case donnée est des mines ou sont tous vides.

Analyse de soustraction

Vérifier les chiffres des deux cases adjacentes et déterminer si nous pouvons signaler ou cliquer en toute sécurité les cases avoisinantes. La logique spécifique est la suivante :



Supposant que les deux cases aux milieux sont les cases chiffrées. On pose et le nombre de mines restant qui doivent être placées dans les cases avoisinantes (le chiffre affiché dans la case moins le nombre de mines avoisinantes déjà déduis). Puis, représentent respectivement le nombre de mines restant dans les zones encadrées. On a évidemment :

Donc,

Si le nombre de case libre dans la zone est égal à , on peut déduire que tous ces cases sont des mines, et toutes les cases libres dans la zone ne sont pas des mines.

Par exemple,

En cours de jouer le démineur, on utilise beaucoup l’analyse de soustraction. Il est la logique la plus fondamentale et la plus importante du démineur. Il existe de divers usages de l’analyse de soustraction, mais on n’a choisi que la situation de deux cases adjacentes parce que c’est le cas plus utilisé est plus facile à appliquer.

Analyse de probabilité

Calculer la probabilité de mine de chaque case par compter tous les jeux possibles. Si la probabilité égale à 0, la case est sécurisée. Tous les cases qui peuvent être déduits par la logique humaines peuvent être déduit par cette analyse, mais on utilise en priorité les deux analyses précédentes parce qu’ils sont beaucoup plus rapides. En plus, s’il n’y a pas des cases à probabilité 0, on va deviner une case parmi les cases avec la plus faible probabilité de mine.

On peut calculer la probabilité de mine une case par :

À cause ça grande complexité en temps, on utilise « diviser pour régner ». D’abord on trouve tous les composants connectés. On définit un composant connecté (CC) comme un ensemble de points, qui indique une zone connectée sur le bord des nombres, dans laquelle les case peuvent interagir les unes avec les autres. Toutes les cases libres ou les cases de question autour d'une case chiffrée sont dans le même CC. Nous mettons donc la première case chiffrée à l'intérieur d'une queue, et chaque fois que nous trouvons une nouvelle case du CC, nous mettons les cases chiffrées avoisinantes de cette nouvelle case dans la queue et faire un BFS pour trouver le CC en entier.

Soit il y a composants connectés en total, on définit le numéro de CC,  
 le numéro de case dans un CC, et le nombre de mine dans un CC.

Pour chaque on trouve toutes les configurations pour des nombre de mines total différents en utilisant le retour arrière (backtracking). On compte le nombre de configurations total , et aussi les nombres des configurations où la case est une mine .

On définit la nombre de configuration de l’ensemble de premiers composants connectés avec le nombre total de mine . On a :

Dans le programme on utilise surtout , la nombre de configuration de l’ensemble de tout les CC sauf . On calcul donc les et par programmation dynamique utilisant la première équation, puis utilisant la deuxième équation.

Il existe encore cases inconnues qui ne sont pas dans les CC. Ces cases sont isolées. La nombre de configuration de ces cases avec mines peut être calculé par combinaison mathématique :

La nombre des configurations de mines de l’ensemble de et les cases isolées :

Finalement, pour tout cases dans un CC, ou bien , sa probabilité de mine est :

Et pour les cases isolées, ses probabilités de mine sont identiques. Selon la théorie des probabilités, la somme de probabilité de toute cases inconnues est égale à . On peut calculer la probabilité de mine de case isolée :