Résolution d'une grille de sudoku :

Class Cell :

Méthode line :

Signature mathématique : $N \rightarrow N$

Calcul asymptotique en O : O (1) -> son sup est O(n)

Axiome : line(idnum) == idnum//9

Peu importe la valeur de self.idnum, cette opération effectue une division entière par 9 ce qui signifie qu'elle aura toujours le même coût, le même temps pour s'exécuter.

Méthode column :

Signature mathématique : $N \rightarrow N$

Calcul asymptotique en O : O (1) -> son sup est O(n)

Axiome : column(idnum) == idnum%9

Peu importe la valeur de self.idnum, cette opération effectue le calcul du reste de la division euclidienne par 9 ce qui signifie qu'elle aura toujours le même coût, le même temps pour s'exécuter.

Axiome line, column : idnum == line(idnum)*9 + column(idnum)

Méthode square :

Signature mathématique : $N \rightarrow N$

Calcul asymptotique en O : O(4) donc son sup est O (n)

Axiome : square(idnum) == self.line( )//3)*3 + (self.column( )//3)

Dans ce calcul, il y a 2 opérations de division entière '//', une multiplication et une addition. Ces opérations ont une complexité constante.

Méthode remove_value :

Signature mathématique : $N \rightarrow B \{0,1\} ou \{True,False\}$

Calcul asymptotique en O : O(1) son sup est O(n)

Il y a une instruction if avec 2 conditions ce qui vaut 1, ensuite si les conditions sont respectées on supprime la valeur du domaine ce qui vaut 1 et on renvoie un booléen ce qui vaut 1.

Axiome : Pour une valeur donnée 'value' et un domaine ' domain' contenant des entiers de 1 à 9, si la case n'est pas verrouillée et que value est dans le domaine alors la value est retiré du domaine. La méthode remove_value renvoie True si le domaine est modifiée.

Si la case est verrouillée ou si value n'est pas dans le domaine aucune modification n'est effectuée et la fonction renvoie False.

Méthode update_value :

Signature mathématique : $N \rightarrow B\{0,1\} ou \{True,False\}$

Calcul asymptotique en O : son sup est O(n)

Étant donné que les opérations effectuées dans cette fonction ont une complexité constante (O(1)), la complexité asymptotique totale reste également constante, même si le nombre d'instructions est de trois. Cela reste en O(1) car le temps d'exécution ne dépend pas de la taille des données ou du domaine, mais reste fixe indépendamment de la taille.

Axiome : Si la case n'est pas bloquée et que son domaine se réduit à une seule valeur, alors la méthode update_value doit assigner cette valeur à l'attribut value de la case, ensuite elle verouille la case en attribuent True à l'attribut locked et renvoyer True pour indiquer que la mise à jour a été faite. Si les conditions ne sont pas remplies, la méthode renvoie False.

Méthode update_domain :

Signature mathématique : $list \rightarrow B\{0,1\} ou \{True,False\}$

Calcul asymptotique en O : son sup est O(n)

Vérification de l'attribut locked, opération en O(1).

Le calcul de l'intersection entre le domaine actuel et la liste de valeurs. La complexité de cette opération dépend de la taille du domaine et de la liste de valeurs. Le domaine a 9 éléments et la liste de valeurs a 9 éléments, la complexité de l'intersection est O(9)

La comparaison entre le domaine mis à jour et le domaine précédent, opération en O(9).

Si la condition est vraie, les opérations suivantes sont exécutées :

self.domain = updated_domain: Mise à jour du domaine, opération en O(1).

return True: indique que le domaine a été modifié, opération en O(1).

Axiome : Pour une liste de valeurs données et un domaine contenant des entiers de 1 à 9, si la case n'est pas verouillée alors le domaine est modifié pour contenir uniquement les valeurs presente à la fois dans le domaine initial et dans la liste de valeurs. Si la case est verouillée aucune modification n'est apporté au domaine initiale.

Méthode reduce_domain :

Signature mathématique : $list \rightarrow B\{0,1\} ou \{True,False\}$

Calcul asymptotique en O : son sup est O(n)

Vérification de l'attribut locked, opération en O(1).

Calcul de la différence entre le domaine actuel et la liste de valeurs. La complexité de cette opération dépend de la taille du domaine et de la liste de valeurs : O(9).

Comparaison entre le domaine mis à jour et le domaine précédent, opération en O(9)

Si la condition est vraie, les opérations suivantes sont exécutées :

Mise à jour du domaine, opération en O(1).

Retourne True pour indiquer que le domaine a été modifié, opération en O(1).

Axiome : Si la case est bloquée, la fonction renvoie False ; si la case n'est pas bloquée et que la valeur est retirée avec succès du domaine, la fonction renvoie True ; si la valeur n'est pas bloquée, la fonction renvoie False.

Class Sudoku :

Méthode reset :

Signature mathématique : $() \rightarrow list$

Calcul asymptotique en O : O (81) son sup est O(n)

La complexité asymptotique de cette fonction est constante.

Méthode __str__ :

Signature mathématique : $() \rightarrow str$

Calcul asymptotique en O : son sup est O (n)

Les opérations et instructions dans cette fonction incluent principalement des conditions et des opérations sur les chaînes de caractères, et leur nombre est proportionnel au nombre de cases dans la grille soit 81.

Méthode line :

Signature mathématique : $\mathbb{N} \rightarrow list$

Calcul asymptotique en O : O (n)

Le nombre d'instructions dans cette fonction dépend directement du nombre de cases dans la grille et de la taille de la ligne sélectionnée. Chaque cellule est évaluée pour sa ligne, donc le nombre d'instructions est proportionnel au nombre de cellules dans la ligne spécifique donc on est sur une complexité constante.

Méthode column :

Signature mathématique : $\mathbb{N} \rightarrow list$

Calcul asymptotique en O : O (n)

Le nombre d'instructions dans cette fonction dépend directement du nombre de cases dans la grille et de la taille de la colonne sélectionnée. Chaque cellule est évaluée pour sa colonne, donc le nombre d'instructions est proportionnel au nombre de cellules dans la colonne spécifique donc on est sur une complexité constante.

Méthode square :

Signature mathématique : $\mathbb{N} \rightarrow list$

Calcul asymptotique en O : son sup est O (n)

Chaque cellule est évaluée pour son carré, donc le nombre d'instructions est proportionnel au nombre de cellules dans le carré spécifique soit 9.

Méthode neighbors :

Signature mathématique : $\mathbb{N} \rightarrow list$

Calcul asymptotique en O : son sup est O (n^2)

La complexité asymptotique de cette fonction dépend du nombre de cellules dans la grille et de la manière dont elles sont vérifiées pour les lignes, colonnes et carrés. Dans le pire des cas, chaque cellule est comparée à chaque autre cellule pour vérifier si elles sont dans la même ligne, colonne ou carré. Ainsi, le nombre d'instructions serait proportionnel au carré du nombre de cellules dans la grille, donc O(n^2), où n est le nombre de cellules.

Méthode propagate :

Signature mathématique : $\mathbb{N} \rightarrow list$

Calcul asymptotique en O : son sup est O (n)

Cette fonction explore les voisins d'une cellule bloquée dans la grille de Sudoku pour supprimer la valeur de la cellule bloquée de leur domaine. La complexité dépend du nombre de cellules voisines dans la même ligne, colonne et carré que la cellule bloquée. Si on considère n comme le nombre total de cellules dans la grille, dans le pire des cas, chaque cellule dans la ligne, colonne et carré de la cellule bloquée est examinée. Ainsi, la complexité serait proportionnelle au nombre total de cellules dans la ligne, la colonne et le carré de la cellule bloquée, soit O(n).

Méthode set_values :

Signature mathématique : $list \rightarrow B\{0,1\} ou \{True,False\}$

Calcul asymptotique en O : son sup est O (n)

Dans le pire des cas, si chaque cellule mise à jour entraîne des mises à jour supplémentaires dans la grille, la complexité pourrait être linéaire par rapport au nombre total de cellules dans la grille, soit O(81).

Méthode grid_parser :

Signature mathématique : $list \rightarrow ()$

Calcul asymptotique en O : O (n)

Méthode find_unique :

Signature mathématique : $list \rightarrow B\{0,1\} ou \{True,False\}$

Calcul asymptotique en O : O (n)

Méthode find_pairs : $list \rightarrow B\{0,1\} ou \{True,False\}$

Signature mathématique :

Calcul asymptotique en O : O (n)

Méthode solve :

Signature mathématique : $\{1,2\} \rightarrow list$

Calcul asymptotique en O : O (n)