Analyse du Problème (Advent of Code – Day 2)

BOULAAJOUL Anass

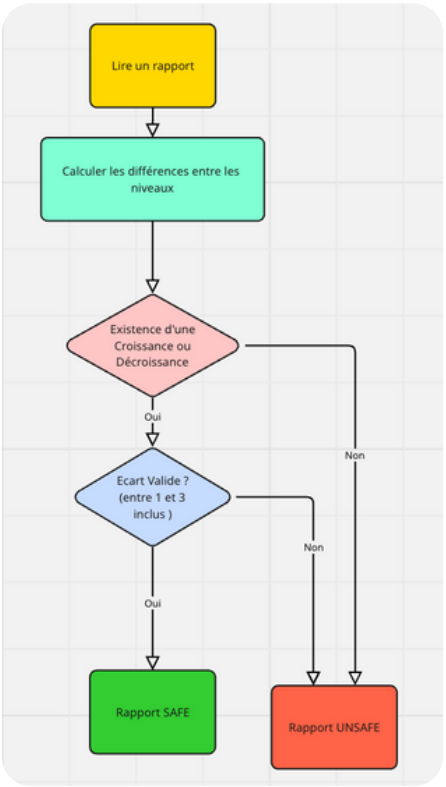
Méthode Utilisée Partie 1:

Méthode STAR

Situation (S)	Tâches (T)	Actions (A)	Résultats (R)
Nous devons analyser un fichier contenant plusieurs rapports numériques afin de déterminer lesquels sont safe ou unsafe selon des règles précises (croissance/décroissance, écart entre 1 et 3).	Comprendre les règles du problème, concevoir une méthode de détection, et produire un programme Java capable de classifier automatiquement tous les rapports.	J'ai rédigé un algorithme simple (lecture → conversion → calcul des différences → vérification), puis j'ai implémenté une solution Java avec gestion du fichier d'entrée et une fonction isSafe() dédiée.	Le programme final lit tout le fichier, analyse chaque rapport et affiche le nombre total de rapports safe, validant ainsi la première partie du puzzle.

Algorithme Utilisée Partie 1:

Algorithme



Méthode Utilisée Partie 2:

Méthode STAR

Situation (S)

Nous devons maintenant analyser les rapports en tenant compte du Problem Dampener, un module permettant d'ignorer une valeur incorrecte dans un rapport pour déterminer si celui-ci peut devenir safe.

Tâches (T)

Adapter la détection précédente afin de vérifier un rapport normal, puis tester si la suppression d'une seule valeur peut rendre un rapport unsafe → safe.

Actions (A)

J'ai ajouté une fonction qui supprime chaque valeur du rapport (une par une), génère une version réduite, et vérifie si cette nouvelle version devient safe. Si l'une d'elles est valide, le rapport est accepté.

Résultats (R)

Le programme mis à jour est maintenant capable d'identifier les rapports safe avec ou sans Dampener, offrant une analyse plus complète et permettant de résoudre la deuxième partie du puzzle.

Algorithme Utilisée Partie 2:

Algorithme