

IFT2015, Été 2023

Travail pratique #2 : Gestion des stocks d'une pharmacie

Deanna Wung, Océane Hays

Autoévaluation

Le programme fonctionne correctement et renvoie des fichiers correspondant à tous les fichiers exemplaires fournis.

Analyse de la complexité temporelle en notation grand O

Variables pertinentes :

k - indique nombre d'items sur la liste de commande;

n - indique nombre de types de médicaments différents;

m - indique nombre d'items sur la prescription;

p - indique nombre (maximal) de médicaments partageant le même nom mais pas la même date d'expiration;

q - représente le nombre de médicaments distincts en stock (donc $q \leq n * p$, parce que $n * p$ va souvent être une surestimation du nombre de médicaments distincts au total)

(voir commentaires dans le code pour plus de détails dans l'analyse)

DATE

- `afficherCommandes()` : Complexité de $O(k)$ car on parcourt la liste de commandes
- `jeterPerimesEtAfficher()` : Complexité : $O(q)$ car on parcourt la totalité des médicaments qu'ils soient du même type ou pas

Complexité globale pour DATE : $O(k + q)$

PRESCRIPTION

- `lirePrescriptions()` : Complexité de $O(p + \log(k) + \log(n))$ pour une seule ligne (un médicament) lu par le programme, car besoin de rechercher le stock (de n éléments), parcourir les p médicaments partageant le même nom, et retrouver l'élément dans la liste de commandes (k éléments).

Complexité globale pour PRESCRIPTION : Avec m médicaments par prescription, on a une complexité de $O(m * (p + \log(k) + \log(n)))$.

APPROV

- `stockerMedicament()` : Complexité de $O(\log(n))$ pour traiter une seule ligne.

Complexité globale pour APPROV : $O(q * \log(n))$, car pour chaque ligne q de APPROV, on a besoin de rechercher et/ou placer dans une liste de n éléments répertoriés en stock (TreeMap).