

Travail pratique 2 : Gestion des stocks d'une pharmacie

Larry Fotso Guiffo - 202201552

Cíntia Dalila Soares - C2791

10 juillet 2023

1 Auto-évaluation

Notre programme fonctionne correctement. On a vérifié les sorties de notre programme avec les sorties attendues dans le fichier *tests* fourni.

2 Analyse de la complexité temporelle en notation grand O

Les complexités temporelles sont exprimées en fonction de :

- n - indique nombre de types de médicaments différents;
- m - indique nombre d'items sur la prescription;
- k - indique nombre d'items sur la liste de demande;

On a utilisé les variables auxiliaires suivantes dans l'analyse dans le code :

- p - indique le nombre des médicaments de même nom, avec des dates d'expirations différentes;
- t - indique le nombre des médicaments avec différents noms;

où $p \leq n$ et $t \leq n$.

L'analyse détaillée a été fait directement dans le code.

2.1 PRESCRIPTION

Puisqu'on lit au maximum m médicaments dans une prescription, et pour chaque prescription lu, on cherche dans le stock s'il y a un médicament avec le même nom $O(\log t)$. Ensuite, on doit vider la file de priorité dans une pile, donc en pire cas on a $O(p)$, pour enfin remettre les éléments de la pile dans la file de priorité $O(p \log p)$. S'il n'y a pas un médicament qui satisfait le médicament de la prescription, on doit ajouter dans la liste de commande, qui est aussi une arbre binaire de recherche, on a donc $O(\log k)$.

On conclut que PRESCRIPTION est $O(mn \log n + m \log k)$.

2.2 APPROV

Puisqu'on lit au maximum n médicaments différents $O(n)$, et que pour chaque médicament lu, on fait soit une insertion dans l'arbre des médicaments, soit une insertion dans la file de priorité, si le médicament était déjà dans l'arbre, ces deux opérations sont $O(\log n)$, on conclut que APPROV est $O(n)$.

2.3 DATE

Puisqu'on doit parcourir toute la liste de commande pour écrire dans le fichier de sortie, on conclut que DATE est $O(k)$.