

Algorithmique et Programmation

TP Comparaison de tris

Fichiers à rendre dans l'archive sur hippocampus : ALGPR_TP5_NOM1_NOM2.zip

```
-- types.h
-- utils.cpp et utils.h
-- tris.cpp et tris.h
-- main.cpp
-- rapport_TP5.pdf
```

0 Présentation du sujet

Une entreprise souhaite organiser sa production d'objets pour limiter les pénalités de retard dues au non-respect des délais de livraison. La production d'un objet dépend donc en premier lieu de la date souhaitée de livraison par le client de l'entreprise, mais aussi de la disponibilité des matières premières nécessaires à sa production.

L'objectif de ce TP est d'adapter un algorithme de tri aux contraintes de livraison de matières premières. Les fichiers fournis sont les suivants :

- les objets à produire avec leurs caractéristiques (**objetsInput**) : nom, nombre, matières premières à la production d'un seul objet (nom, poids nécessaire (kg)), date livraison, durée de production
(un lot d'objets doit être traité comme un bloc et ne peut pas être séparé)
- livraison matières premières (**matieres**) : identifiant livraison, nom, poids livraison (kg), date arrivée

1 Structure de données

Définissez dans le fichier **types.h** les structures de données permettant de stocker en mémoire les éléments suivants :

- un objet à produire **objet** et un ensemble d'objet **ensObjets**
- une matière première **matiere** (**nom**, **poids**) et un ensemble de matières premières **ensMatiere**
- une livraison de matière première **livraison** et un ensemble de livraison **ensLivraison**

Utilisez **vector<>** pour définir les ensembles. Ajoutez dans ce fichier les constantes et les autres types dont vous pourriez éventuellement avoir besoin.

2 Lecture et écriture

- Déclarez et définissez, respectivement dans les fichiers **utils.h** et **utils.cpp**, les fonctions suivantes :
 - **lireObjets** pour lire les objets dans le fichier **objetsInput_X.txt**
format par ligne : **nom nombre nombre_matiere_premieres nom_mat1**
poids_mat1 ... nom_matN poids_matN date_souhaitée duree_production_lot

Les matières premières sont données pour un objet. La durée de production est donnée pour le lot d'objet.

— `ecrireObjets` pour écrire les objets triés dans le fichier `objetsOutput_X.txt`

Remarque : vous pouvez utiliser la méthode `eof()` de `fstream` pour détecter la fin d'un fichier.

- Testez ces fonctions dans un programme principal (`main.cpp`) pour comparer les temps d'exécution sur les jeux de données fournis. Indiquez les résultats de tests dans votre rapport.

3 Tris selon la date lors de la lecture

- Modifiez la fonction de lecture pour trier les éléments au fur et à mesure de leur lecture. Proposez deux versions pour rechercher la position de l'élément à insérer 1/ recherche linéaire 2/ recherche dichotomique. Le critère de tri est la date de livraison souhaitée.

Remarque : Pour les dates, vous devez utiliser une référence temporelle absolue, à savoir le nombre de jours depuis l'ouverture de l'appareil de production, initialisé à 0 au début de l'algorithme principal.

- Testez ces fonctions dans le programme principal (`main.cpp`) pour comparer les temps d'exécution sur les jeux de données fournis, et les mettre en relation avec la complexité des algorithmes. Indiquez les résultats de tests dans votre rapport avec un tableau et un graphique.

4 Calcul du retard

- Déclarez et définissez une fonction `calculRetard` pour calculer le nombre de jours de retard et les pénalités dans les fichiers `utils.h` et `utils.cpp`. Le coût des pénalités de retard est de 1000 euros par jour par objet. Elle prend les objets triés en entrée et retourne deux valeurs en sortie.
- Testez cette fonction dans le programme principal (`main.cpp`) sur les jeux de données fournis. Indiquez vos résultats de tests dans votre rapport.

5 Matières premières

- Déclarez et définissez, respectivement dans les fichiers `utils.h` et `utils.cpp`, la fonction suivante :
 - `lireMatiere` pour lire les matières premières dans le fichier `matieres_X.txt` format par ligne : `id_livraison nom_matiere poids_livraison date_livraison`
- Modifiez la fonction `calculRetard` pour tenir compte de la disponibilité des matières premières pour lancer la production d'un objet.
- Testez cette fonction de la même manière que la question précédente et donnez de la même manière vos résultats et conclusions. Vous démarrez au `t0` avec un stock de 2 tonnes pour chaque matière première.

6 Rapport

Précisez dans votre rapport, les structures de données finales, toutes les spécifications des fonctions utilisées ainsi que la **dernière version implémentée des algorithmes des fonctions de lec-**

ture/tri et de calcul du retard. Vous ajouterez les tests des algorithmes des questions 3, 4, 5 (temps exécution et coûts).