

Programmation efficace – Examen de session 1, 2022-2023

F. Laroussinie, A. Sangnier

25 mai 2023, 14h00-17h00

L'objectif est de gérer des réservations pour une salle de réunion avec différentes tables. On dispose pour cela de Ta ($Ta \geq 1$) tables de différentes tailles et de plusieurs demandes de réservation. Une demande de réservation contient les informations suivantes :

1. P un entier correspondant à un nombre de personnes (avec $0 < P < 1000$) ;
2. T un temps (en minutes) disant l'heure de la réservation (avec $0 \leq T < 100000$) ;
3. M une marge disponible en minutes par rapport à l'heure de la réservation (avec $0 \leq M < 1000$) ;
4. D la durée de la réunion en minutes (avec $0 < D < 1000$).

On suppose de plus qu'il y a une date limite (en minutes), notée L , au delà de laquelle il n'est plus possible de travailler.

L'objectif est de choisir le meilleur (le critère est expliqué ci-dessous) ensemble de **réservations placées compatibles**. Une **réservation placée** est une réservation pour laquelle on a fixé une date de début t_{deb} (en respectant les valeurs T et M de la réservation, c'est-à-dire avec $T \leq t_{deb} \leq T + M$ et en s'assurant que la réunion ne termine pas après la date limite L) et à laquelle on a associé une table i (avec $0 \leq i < Ta$). Ces réservations placées doivent être **compatibles** au sens où à tout moment, pour chaque table, il y a assez de places pour accueillir tous les participant-e-s des réunions choisies pour cette table et qui se déroule à la date donnée.

Une solution au problème sera donc un ensemble de triplets (num, t_{deb}, i) : un numéro de réservation, sa date de début et la table où se tiendra cette réunion. Notons que si cette réservation a une durée D , cela signifie qu'entre les dates t_{deb} et $t_{deb} + D - 1$ (inclus), P places de la table i sont occupées par cette réunion.

L'objectif est d'accepter le plus de personnes possibles pour les réunions. C'est le critère d'optimalité choisi.

Données d'entrée

Les données d'entrées sont décrites dans un fichier qui a la forme suivante :

- sur la première ligne, on trouve trois entiers Ta , N et L (strictement positifs) séparés par des espaces tels que :
 - Ta représente le nombre de tables (numérotées de 0 à $(Ta - 1)$) ;
 - N représente le nombre de réservations (numérotées de 0 à $(N - 1)$) ;
 - L représente la date limite jusqu'à laquelle des réservations sont autorisées (on pourra occuper des tables entre les instants 0 et $L - 1$).
- les lignes numérotées de 2 à $(Ta + 1)$ contiennent le nombre de places de chacune des tables, c'est-à-dire qu'elles n'ont qu'un seul entier strictement positif ;

- les lignes numérotées de $Ta+2$ à $(Ta+N+1)$ contiennent les données de chaque demande de réservation. Elles sont composées de 4 entiers strictement positifs séparés par des espaces et représentant dans l'ordre les données P , T , M et D d'une réservation.

```

3 8 200
1
3
5
2 0 10 20
1 0 50 100
4 20 0 10
3 10 40 60
5 0 0 200
3 100 0 20
3 100 20 70
2 100 10 80

```

FIGURE 1 – Fichier d'entrée exemple.in

La figure 1 fournit un exemple de fichier d'entrée avec 3 tables et 8 demandes de réservation et les réservations doivent être placées entre le temps 0 et le temps 199. La table 0 dispose de 1 place, la table 1 a trois places et la table 2 a 5 places. Il y a de plus 8 demandes de réservation. Par exemple, la demande 0 est pour 2 personnes, elle doit commencer entre le temps 0 et le temps 10 et la réunion durera 20 minutes. La réservation 2 est pour 4 personnes, elle doit commencer au temps 20 et elle durera 10 minutes. Quant à la réservation 7, elle est pour 2 personnes, elle doit commencer entre le temps 100 et le temps 110 et elle durera 80 minutes.

Données de sortie

La donnée de sortie est un fichier décrivant les réservations acceptées, leur date de début et leur placement. La première ligne du fichier contient un entier positif décrivant le nombre de réservations acceptées. Ensuite pour chaque réservation acceptée, on donnera sur une ligne trois entiers correspond à son numéro, sa date de début (en minute) et sa table. Ces trois données étant séparées par des espaces. Et les réservation acceptées doivent respecter les contraintes suivantes :

- Toutes les personnes d'une réservation sont assises à la table pendant toute la durée de la réunion.
- Il n'y a jamais plus de personnes à une table que la taille de la table..
- Toute réunion commence nécessairement entre l'heure T et l'heure $T + M$ (si P , T , M et D sont les données de la réservation).
- La réservation dure forcément D minutes.
- Chaque réservation n'est présente qu'au plus une fois dans le fichier de sortie (et peut bien entendu ne pas être présente).

Tous les fichiers de sortie présentés ici correspondent au fichier d'entrée donné à la Figure 1. Par exemple, sur la figure 2, nous avons donné un fichier de sortie valide avec les caractéristiques suivantes :

- Il y a 7 réservations placées.

```
7
0 0 1
1 10 0
2 20 2
3 20 1
5 100 2
6 120 2
7 100 2
```

FIGURE 2 – Fichier de sortie exemple1.out

- La réservation 0 commence au temps 0 à la table 1. Comme cette réservation dure 20 minutes et qu'elle est pour deux personnes, alors elle occupera 2 places de la table 1 entre les temps 0 et 19 inclus.
- La réservation 1 commence au temps 10 à la table 0. Comme la réservation 1 devait commencer entre le temps 0 et 50, ces données sont correctes. Cette réservation occupera donc 1 place de la table 0 entre les temps 10 et 109.
- La réservation 2 commence au temps 20 à la table 2. Cette réservation occupera donc 4 places de la table 2 entre les temps 20 et 29.
- La réservation 3 commence au temps 20 à la table 1. Cette réservation occupera donc 3 places de la table 1 entre les temps 20 et 79.
- La réservation 5 commence au temps 100 à la table 2. Cette réservation occupera donc 3 places de la table 2 entre les temps 100 et 119.
- La réservation 6 commence au temps 120 à la table 2. Cette réservation occupera donc 3 places de la table 2 entre les temps 120 et 189.
- La réservation 7 commence au temps 100 à la table 2. Cette réservation occupera donc 2 places de la table 2 entre les temps 100 et 179.

```
4
0 0 1
1 10 0
3 20 1
4 0 2
```

FIGURE 3 – Fichier de sortie exemple2.out

Sur la figure 3, nous avons donné un autre exemple de fichier de sortie valide avec la différence que la table 2 est désormais entièrement occupée par la réservation 4 entre les temps 0 et 199.

La figure 4 donne un exemple de fichier incorrect. En effet, sur ce fichier la réservation 0 occupe 2 places de la table 1 entre les temps 0 et 19 inclus et la réservation 3 occupe 3 places de la table 1 entre les temps 10 et 69 inclus, il y a donc un moment où 5 personnes sont à la table 1 qui est de taille 3.

```
2
0 0 1
3 10 1
```

FIGURE 4 – Fichier de sortie **incorrect** exemple3.out

Calcul des points

Nous allons maintenant expliquer le calcul des points pour un fichier de sortie valide. Il s'agit tout simplement du nombre de personnes qui ont pu faire une réunion.

Pour les exemples correspondants au fichier d'entrée de la Figure 1, on a :

- pour le fichier de la Figure 2, le score est de 18;
- pour le fichier de la Figure 3, le score est de 11.

Nous vous fournissons un fichier JAR `JAVA TestExam23.jar` qui doit être appelé de la façon suivante : `java -jar TestExam23.jar entree.in sortie.out` et qui dit si le fichier `sortie.out` est un fichier de sortie valide pour le fichier d'entrée `entree.in` ou non et le cas échéant affiche le score.

Différents exemples de fichiers d'entrée sont aussi fournis sur cette page web.

Rendu

Le but de l'examen est de fournir un ou des programmes fabriquant des fichiers de sortie valides pour les fichiers d'entrée donnés sur la page web :

<https://www.irif.fr/sangnier/enseignement/prog-eff.html>,

le but est aussi d'obtenir des scores les plus élevés possibles.

Vos programmes peuvent être en Java, Python ou C!

À la fin de l'examen, vous devez envoyer vos solutions dans une archive zip ou tar à `francoisl@irif.fr` et `sangnier@irif.fr`. Cette archive contiendra vos programmes produisant des fichiers de sortie et un fichier `readme.txt` expliquant ce que vous avez fait et comment utiliser vos programmes et **rappelant vos noms**.