

Devoir maison

Recherche opérationnelle S3.

Zeyu CHEN

Numéro d'étudiant : 20170019

Ce devoir maison a été généré automatiquement et aléatoirement pour Zeyu CHEN. Il contient 3 exercices : un exercice d'application du cours sur 1 point, un exercice intermédiaire sur 1 point et un algorithme à coder sur 2 points. Pour répondre à ce devoir, vous devez rendre, sur `exam.ensie.fr`, un fichier texte nommé `chenzeyu_20170019.txt`. Ce fichier devra contenir **dans l'ordre** les solutions des 3 exercices tels que demandé dans chaque exercice. Toute ligne vide dans le fichier sera ignorée (vous pouvez donc sauter des lignes entre 2 exercices, ou au sein d'un exercice). **ATTENTION** : un non respect du format pour un exercice entraînera la note de 0 pour l'exercice. Il vous est possible de tester votre format ainsi que votre dernier exercice avec le script `check` qui vous a été fourni avec ce sujet. Vous pouvez par exemple le copier avec votre fichier solution sur une machine de l'école et exécuter

```
./check chenzeyu_20170019.txt
```

Si votre fichier est au bon format, le script vous le dira explicitement. Le code de votre dernier exercice peut être dans ce cas testé sur 100 tests unitaires. Si un test ne passe pas, le script vous l'affichera pour que vous puissiez corriger votre code. Vous pouvez demander de l'aide aux professeurs/chargés de TD, ou aux autres élèves, mais sachez que le sujet est différent pour tout le monde.

Exercice — #1-7-1 Probabilités d'une file d'attente

On considère une file d'attente à 1 guichet avec un taux de naissance constant $\lambda = 48$ et un taux de mort constant $\mu = 76$. Calculez la probabilité P_0 qu'il y ait 0 personne dans la file en régime permanent, si ce régime existe.

Vous donnerez cette réponse au format suivant : vous noterez en premier le numéro de l'exercice précédé par un #, puis une ligne contenant un flottant égal à la probabilité P_0 si le régime permanent existe. Ce nombre doit être écrit avec la notation scientifique, arrondi 2 chiffres après la virgule. Si le régime permanent n'existe pas alors la ligne doit contenir à la place le nombre -1.

Par exemple :

```
# 1-7-1
```

```
3.80E-1
```

Exercice — #2-5-2 Solution optimale

Soit le programme linéaire suivant :

$$\left\{ \begin{array}{lll} \min & -2x & -y \\ & 3x & -7y \leq 13 \\ & x & \leq 2 \\ & -x & \leq 5 \\ & x & +7y \leq 16 \\ & x, & y \in \mathbb{R} \end{array} \right.$$

Déterminez une solution optimale du programme linéaire. S'il en existe plusieurs, donnez celle qui minimise x . Si cela ne départage toujours pas les solutions, donnez celle qui minimise y . La solution demandée est nécessairement à coordonnées entières.

Vous donnerez cette réponse au format suivant : vous noterez en premier le numéro de l'exercice précédé par un #, puis 1 ligne contenant 2 entiers x et y séparés par une espace où (x, y) est la solution optimale demandée.

Par exemple :

```
# 2-5-2
```

```
-2 1
```

Exercice — #3-6-2 Algorithme de calcul des classes d'états communicants

Coder en C un algorithme calculant les classes d'état communicants d'une chaîne de Markov.

Pour cela, vous coderez une fonction avec la signature suivante :

```
void communicating_classes_algorithm(int n, float** transitions, int* comclasses);
```

- `n` est le nombre d'états de la chaîne de Markov numérotés de 1 à n .
- `transitions` est un tableau de $n \times n$ flottants où `transitions[i-1][j-1]` est la probabilité de transition de l'état i vers l'état j .
- `comclasses` est un tableau de `n` entiers. En entrée, `comclasses[i-1]` vaut -1 pour tout $1 \leq i \leq n$. Votre fonction doit affecter dans `comclasses[i-1]` un entier entre 1 et n , quel qu'il soit, tel que, si deux états i et j sont dans la même classe d'états communicants si et seulement si `comclasses[i-1] = comclasses[j-1]`.

On garantit, que, quelque soit l'entrée $2 \leq n \leq 100$.

Votre fonction doit compiler avec `gcc -Wall -Wextra -ansi` et se terminer en moins de 1 seconde. Vous pouvez déclarer d'autres fonctions ou des structures mais ne pouvez ni écrire de `#define` ou de `#include`. Vous pouvez utiliser tout ce qui est défini dans `stdlib.h` à l'exception de `system`. Cette bibliothèque sera incluse automatiquement. Ne mettez pas de commentaire. Elle sera testée sur 100 cas. Votre note est de 2 si tous les tests sont réussis, 1 si plus de 10 pourcents mais pas tous, et 0 sinon. **Conseil** : n'utilisez pas de tableaux, utilisez des pointeurs à la place.

Vous donnerez cette réponse au format suivant : vous noterez en premier le numéro de l'exercice précédé par un #, puis écrivez votre code.

Par exemple :

```
# 3-6-2
void communicating_classes_algorithm(int n, float** transitions, int* comclasses){
}
```