

Diviseur de classes COCOAIRLINES

La compagnie aérienne COCOAIRLINES souhaite équiper sa flotte d'avions de diviseur de classes permettant ainsi des configurations et des tailles de classes différentes. La compagnie propose des sièges dans des classes de type *First*, *Business* et *Economy*. Elle propose également la location des sièges à d'autres compagnies aériennes.

Les diviseurs étant très cher et occupant un espace non négligeable dans un avion, l'objectif est de positionner n diviseurs dans un avion qui a une capacité de m blocs de sièges. Les diviseurs ne peuvent pas être installés au niveau des sorties de secours *exits* (au moins une). Il est aussi important d'avoir la classe à l'avant de l'avion d'une taille supérieure ou égale à deux blocs. Pour assurer des configurations avec des tailles de classes différentes, il suffit de s'assurer que les distances entre les diviseurs sont différentes. La figure 1 illustre le problème avec un avion d'une capacité $m = 11$ blocs, $n = 5$ diviseurs à installer, une seule sortie de secours à $exists_1 = 3$.

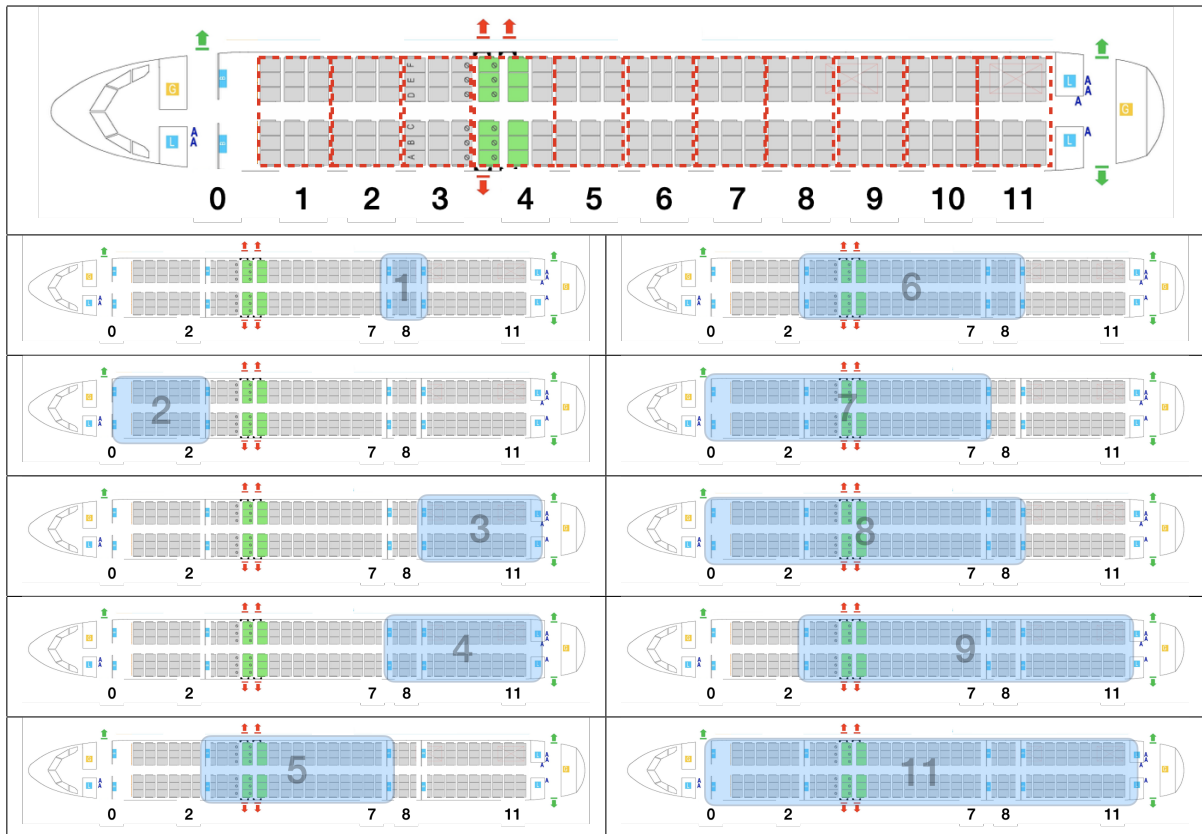


Figure 1: Exemple d'avion avec une capacité de 11 blocs, 5 diviseurs à installer et une issue de secours.

Question 1 • Proposez un algorithme et son implémentation Java pour résoudre le problème des diviseurs de classes $dividers(n, m, exits)$.

Question 2 • Testez votre solution sur les instances du dépôt.

Question 3 • Modélisez maintenant $dividers(n, m, exits)$ sous la forme d'un réseau de contraintes.

Question 4 • Implémentez votre réseau de contraintes sous choco à partir des fichiers du dépôt github.

Question 5 • Testez la version choco sur les instances du dépôt et comparez avec les solutions de l'enseignant sur le dashboard.

Question 6 • Chargez votre réalisation choco sous la forme d'un jar exécutable dans la plateforme (CP-Dashboard). Le jar doit avoir uniquement l'affichage de `printStatistics()` sur la console. Vous avez accès aux stats des solutions de l'enseignant et des autres équipes pour une comparaison en temps réel.

Question 7 • Optimisez votre modèle en expliquant dans le détail l'optimisation apportée (ex. contraintes globales, élimination des symétries, contraintes redondantes...).