

# Devoir à la maison

Vous travaillez dans une entreprise de logistique pour les fêtes et on vous donne  $n$  objets de poids  $p_1, \dots, p_n$  à expédier. Vous avez à votre disposition des colis, qui peuvent chacun contenir un poids au plus  $P$ . Votre but est de ranger les objets dans les colis, en utilisant le moins de colis possible et en respectant la contrainte que la somme des poids d'un objet d'un colis est inférieure ou égale à  $P$ . On appelle ce problème le problème du *remplissage des lutins*. Une solution du problème du remplissage des lutins est donc une liste de donc chaque élément est un colis constitué de la liste des objets qui le remplit. Par exemple, si vos colis font un poids de 10 kilos maximum et que les objets que vous devez ranger font  $(1, 4, 1, 6, 7, 9, 3, 2, 8)$  alors  $(1, 1, 2, 6), (3, 7), (4), (8), (9)$  est une solution utilisant 5 colis. Une solution est dite *optimale* si elle utilise un nombre minimal de colis parmi toutes les solutions.

## 1 Partie théorique

1. Que peut-on toujours faire si on a deux objets de poids  $p_1$  et  $p_2$  tel que  $P = p_1 + p_2$ .
2. Que peut-on dire si un algorithme remplit deux colis à moins de la moitié de leur capacité ?
3. Proposer au moins deux algorithmes gloutons qui résolvent le problème du remplissage des lutins (mais pas nécessairement en donnant la solution optimale).
4. Proposer également un algorithme qui trouve toujours la solution optimale (mais qui peut être lent).
5. Faites tourner vos algorithmes sur l'exemple :  $(2, 6, 1, 5, 8, 4, 5, 7, 5, 3)$  et  $P = 9$  et donner la réponse obtenue.
6. Donner pour chacun de vos algorithmes gloutons une entrée pour laquelle il trouve une solution optimale.
7. Le facteur d'approximation d'un algorithme est le pire rapport, sur toutes les entrées possibles, entre le nombre de colis trouvé par l'algorithme divisé par le nombre de colis optimal. Pour un algorithme qui trouve la solution optimale, ce facteur sera toujours un. Donner une borne sur le facteur d'approximation d'un de vos algorithmes glouton.

L'étudiant qui aura trouvé le meilleur facteur d'approximation aura des points bonus.

## 2 Partie pratique

Le code est à rendre avec les réponses à la partie précédente. Il peut être fait dans le langage de votre choix.

1. Implémentez les algorithmes de la partie précédente.
2. Implémentez un algorithme qui génère une entrée aléatoire avec  $n$  objets dont le poids est tiré aléatoirement entre 1 et  $m$ .

3. Pour chacun de vos algorithmes, calculer le nombre moyen de colis nécessaires, pour 1000 entrées avec 100 objets de poids entre 1 et 10 et un poids maximal par colis de 10. Vous calculerez également la moyenne sur 1000 entrées avec 100 objets de poids entre 1 et 1000 et un poids maximal par colis de 1000.

L'étudiant dont l'algorithme aura la meilleure performance moyenne calculée à la question précédente aura des points bonus.