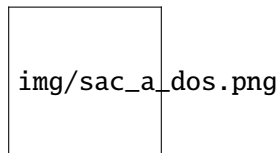


## Exercice : Problème du sac à dos

---

Dans un jeu vidéo, le héros dispose d'un sac à dos lui permettant de porter les objets collectés au cours de sa quête. Le sac à dos a une capacité maximale de 10 kg. Le héros doit maximiser la valeur en pièces d'or des objets contenus dans son sac à dos. Chaque objet a une masse en kg et a une valeur en pièces d'or.

On donne la valeur et la masse des différents objets collectés par le héros.



- 1) Afin de remplir le sac, on pourrait tester toutes les possibilités et choisir la meilleure d'entre elles. Expliquer comment cet algorithme, qualifié de « naïf », permettrait de trouver la ou les solution(s) optimale(s).
- 2) En observant que le choix de chaque objet est binaire (choisi ou non), combien de possibilités différentes pourrait-on dénombrer avec 7 objets possibles ? Et avec 100 objets ? Conclure sur la durée d'exécution d'un tel algorithme naïf.
- 3) Un algorithme glouton pourrait également être utilisé. Préciser la différence majeure entre l'algorithme naïf et l'algorithme glouton.
- 4) À choisir entre la baguette magique et la cape d'invisibilité, quel choix conseillerez-vous à votre héros et pourquoi ?
- 5) Parmi les critères suivants, cocher le(s) plus pertinent(s).
  - ☐ Choisir en premier l'objet le plus lourd / le moins lourd.
  - ☐ Choisir en premier l'objet le plus précieux / le moins précieux.
  - ☐ Choisir en premier l'objet ayant la valeur massique (en pièces/kg) la plus élevée / la moins élevée.
  - ☐ Prendre le plus d'objets possible dans le sac.
- 6) Calculer la valeur massique de chaque objet en pièces d'or par kilogramme.
- 7) Quel pourrait être le meilleur choix possible à chaque étape d'ajout d'un objet dans le sac ?
- 8) Quel pré-traitement faudrait-il effectuer avant de remplir le sac à dos ? Quels objets seraient alors choisis, et dans quel ordre ?
- 9) Déterminer la composition du sac à dos en indiquant les objets choisis, leur ordre de choix ainsi que la valeur en pièces d'or et la masse en kilogramme du sac à dos.
- 10) Écrire, en Python ou en pseudo-code, un algorithme glouton répondant au problème posé.
- 11) Quelle est la complexité de cet algorithme en supposant que le tri a une complexité d'ordre  $O(n \log(n))$  ?
- 12) Comparée la solution trouvée avec la solution suivante : { Baguette, Cape, Diadème, Miroir }. Conclure sur le caractère optimal de l'algorithme glouton.