

TD 2



Sablier

Xavier Pessoles

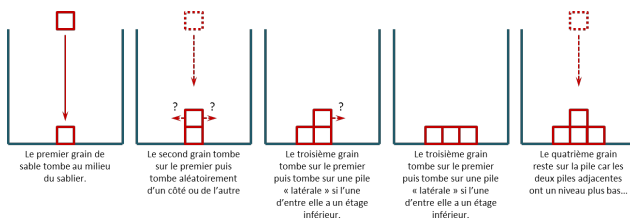
Savoirs et compétences :

- Alg – C16 : Piles - Algorithmes de manipulation : fonctions «push» et «pop».

Exercice 1 - Le marchand de sable

On se propose de modéliser la constitution d'un tas de sable ainsi que l'écoulement des grains dans un sablier. Afin de simplifier le problème, on se restreindra à travailler en 2 dimensions. Le tas sera modélisé par une pile de grains de sable.

Dans le cas du sablier, les grains tombent toujours sur la même pile. Le processus de constitution de la pile est le suivant :



Gestion d'une pile de sable

Objectif Modéliser l'écoulement d'un grain de sable sur une colonne.

Une pile de sable est modélisée par... une pile! Cette dernière est implémentée sous forme d'une liste. La taille des piles est de dimension finie notée `ht`. Une pile de 3 grains de sable dans une pile de taille 5 sera modélisée par la liste `['*', '*', '*', ' ', ' ']`.

Question 1 Donner l'implémentation des fonctions élémentaires permettant de gérer une pile dans Python à savoir les fonctions `creer_pile`, `empiler`, `depiler`, `est_vide`. Pour cette question on s'autorise l'utilisation des méthodes sur les listes. Vérifier le bon comportement de vos fonctions.

Question 2 Donner l'implémentation de la fonction `taille_pile`, permettant de connaître la taille d'une pile. Pour cette question seules les fonctions définies ci-dessus sont acceptées. Donner la complexité algorithmique de la fonction implémentée. Vérifier le bon comportement de vos fonctions.

Question 3 Implémenter une fonction `simulation_pile` prenant comme arguments la hauteur de la pile et le nombre de grains à faire tomber. Cette fonction devra retourner une liste de piles de grain de sables. Cette liste stockera donc les différents états de la pile au fur et à mesure de la chute des grains.

Question 4 Utiliser la fonction `trace_ecoulement` pour tracer l'écoulement des grains sur une pile.

Question 5 Redéfinir la fonction `empiler` en la nommant `empilerSable` pour que le seul élément empilable soit la chaîne de caractères `"*"`. Ainsi, une pile de sable sera constituée d'une pile d'étoiles.

Gestion du tas de sable

Objectif L'objectif est de modéliser les fonctions élémentaires afin de pouvoir manipuler un tas de sable et pas une colonne de sable.

Un tas de sable est maintenant modélisé par une **liste de piles** de grains de sable. Pour cette liste, les méthodes de liste peuvent être utilisées.

Question 6 Implémenter la fonction `creation_sablier` d'arguments `larg` et `haut` permettant de créer un sablier constitué de `larg` piles de hauteur `haut`.

Question 7 Modifier les fonctions `empiler` et `depiler` de telles sortes qu'elles prennent comme argument un sablier et un numéro de colonne `col`. On pourra ainsi déposer ou supprimer un grain de sable sur une colonne `col`.

Modélisation de la chute d'un grain

Objectif On souhaite commencer par simuler la chute libre d'un seul grain.

Question 8 Implémenter la fonction `chute_libre_grain` d'arguments `sablier` (sablier quelconque), `col` (numéro de la colonne ou le grain chute), `simu` (liste de sabliers permettant de visualiser la simulation). Cette fonction

permettra de stocker dans la variable `simu` l'ensemble des états de la variable `sabliers`. Ains, un tracer de la variable `simu` permettra de visualiser la chute d'un grain.

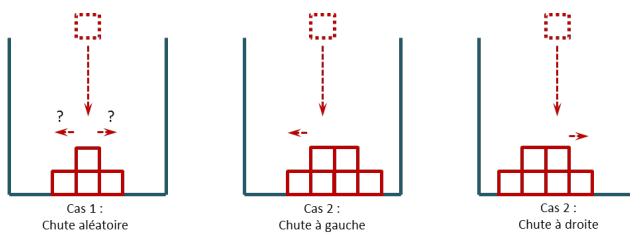
Question 9 Donner les instructions permettant de dépiler un grain de sable sur la pile i et d'empiler un grain de sable sur la pile $i - 1$ du tas à n piles. Tester les instructions pour la pile $i = 2$ du `tasSable = [[], ["*"], ["*", "*", "*"], ["*"], ["*",`

Écoulement

On va maintenant implémenter les fonctions qui vont permettre de régir l'écoulement d'un grain de sable. On suppose que les grains tombent toujours sur la même pile.

On s'intéresse d'abord au sens d'écoulement d'un grain de sable. Pour cela, on définit une variable `sens` qui vaut 0 lorsque le grain doit s'écouler vers la gauche et qui vaut 1 lorsque le grain doit s'écouler vers la droite.

On identifie les 3 cas suivants pour déterminer le sens de chute d'un grain :



On compare la taille des piles avant que le grain de sable soit tombé.

Question 10 Exprimer la condition booléenne pour laquelle un grain de sable chute à gauche.

Question 11 Exprimer la condition booléenne pour laquelle un grain de sable chute aléatoirement à gauche ou à droite.

Question 12 En réalisant un schéma, donner un cas de figure pour lequel il n'y a pas d'écoulement de grain. Traduire la condition booléenne correspondante.

Question 13 Exprimer la condition booléenne permettant de savoir si un grain qui tomberait sur la pile n doit s'écouler sur la gauche. On tiendra compte du cas où le grain est sur le bord du sablier.

Question 14 Implémenter la fonction `sens` permettant de déterminer le sens de la chute du grain : droite, gauche ou si le grain tombe sur sa pile. Cette fonction prendra comme arguments `tas(liste de piles)` et `indice(int)` l'indice de la pile sur laquelle le grain sera lâché.

On appelle `chute` la fonction permettant de régir la chute du grain. Les spécifications de la fonction sont les suivantes :

Python

```
def chute(tas, indice):
    """
    Gestion d'une chute de grain de sable.
    Entrées :
    * tas(liste de piles) : tas de sable
    * indice(int) : pile sur laquelle le dernier
      grain de sable va tomber
    * sens(int) : 0 chute à gauche, 1 chute à
      droite
    None si le grain reste sur la pile
    Sortie :
    * le tas est modifié mais n'est pas retourné.
    """
```

Question 15 Implémenter la fonction `chute` permettant de gérer la chute d'un grain de sable. Cette fonction devra être récursive. Tester la fonction `chute` à partir d'un tas de sable de 7 piles sur lequel vous ferez chuter 12 grains de sable sur la pile d'indice 3.

Affichage du tas de sable

On donne le tas suivant :

Python

```
>> print(tas)
[[], ['*'], ['*', '*'], ['*', '*', '*'],
 ['*', '*'], ['*'], []]
```

On souhaite l'afficher sous la forme suivante :

Python

```
-----
---*---
---*---
*****
```

Question 16 Implémenter la fonction `affichage` permettant d'afficher un tas sous la forme définie ci-dessus. Tester l'affichage du tas créé à la question précédente.