Aix*Marseille université
Initiative d'excellence
Année 2018-19

Diplôme Inter-Universitaire
Enseigner l'informatique au lycée

**Bloc 1 : TD 5**

**Correction d'algorithme et introduction à la complexité en mode d**

# 1 Correction du tri par sélection

Etant donné un tableau de n entiers `T[0], T[1], ..., T[n-1]` le principe du tri par sélection est le suivant.

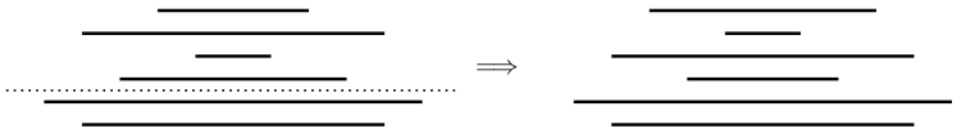
On suppose que la première partie du tableau contient les `k` plus petits éléments triés. Au départ `k=0`. On choisit l'élément minimal parmi ceux qui ne sont pas encore triés (les éléments de rang `k` à `n-1`) et on échange sa position avec celle de l'élément de rang `k`. On incrémente `k` et on recommence tant que `k<n`.

**Question 1.** Décrire en Python ou en pseudo-code un algorithme de recherche de l'indice de l'élément minimal parmi ceux de rang `k` à `n-1`. Prouver qu'il est correct.

**Question 2.** Décrire en Python ou en pseudo-code un algorithme de tri par sélection. Prouver qu'il est correct.

# 2 Crêpier psychorigide (en mode débranché)

Dans une crêperie du Vieux Port, deux crêpiers se relaient en cuisine. Toutes les crêpes n'ont pas exactement le même diamètre. Au moment du changement de crêpier, s'il reste des crêpes déjà précuites, le crêpier sur le départ, un peu psychorigide sur les bords, veut laisser à son collègue une pile de crêpes triée de la plus grande en bas, jusqu'à la plus petite en haut. Oui, mais voilà : la cuisine est minuscule ! La seule possibilité pour le crêpier psychorigide est d'utiliser sa spatule, de la planter entre deux crêpes de la pile de crêpes et de retourner la totalité de la pile de crêpes au-dessus, le tout sur la même pile de crêpe. Par exemple, dans la pile de crêpes à gauche ci-dessous, si le crêpier plante sa spatule à l'endroit des pointillés et retourne la pile du dessus, il obtient la pile de crêpes de droite.



**Question 1.** Comment le crêpier doit-il s'y prendre pour trier sa pile de crêpes ? Décrire une méthode que le crêpier peut utiliser facilement : en particulier, notez qu'il est facile pour le crêpier de trouver le plus grande crêpe dans une pile de crêpes...

**Question 2.** Décrire votre algorithme (en pseudo-code, pas nécessairement en Python...) en supposant que la pile de crêpes est représentée par un tableau `pile_crepes` contenant les diamètres des `n` crêpes en commençant par la crêpe la plus basse. Vous pourrez utiliser

- la fonction `retourner_spatule(pile_crepes, i)` pour dire au crêpier de planter sa spatule au-dessus de la `i`-ème crêpe du tas `pile_crepes` et de retourner la pile au-dessus (l'exemple du début correspond donc à `retourner_spatule(pile_crepes, 2)`) ;
- la fonction `plus_grande_crêpe(pile_crepes, i)` qui retourne la plus grande crêpe dans le tas `t` au-dessus de la `i`-ème crêpe, c'est-à-dire dans le sous-tableau `pile_crepes[i..n-1]` : ainsi `plus_grande_crêpe(pile_crepes, 0)` renvoie la plus grande crêpe du tas, alors que `plus_grande_crêpe(pile_crepes, 1)` fait de même en ignorant la crêpe du bas.

**Question 3.** Combien d'opérations élémentaires (recherche de la plus grande crêpe dans un sous-tas et retournement d'un sous-tas) effectue le crêpier s'il utilise votre algorithme dans le pire des cas, en fonction du nombre $n$ de crêpes dans la pile ?

**Question 4.** Proposer des algorithmes (en pseudo-code ou en Python) pour réaliser les fonctions `plus_grande_crêpe` et `retourner_spatule`, en ne s'autorisant comme opérations élémentaires sur les tableaux

que la lecture et l'écriture dans une case du tableau.

**Question 5.** En supposant désormais qu'on compte comme opérations élémentaires les comparaisons d'éléments du tableau et les lectures et écritures dans le tableau, quelle est la complexité du tri du crêpier psychorigide ?