

DS 1

Informatique pour tous, deuxième année

Julien REICHERT

Durée : 2 heures maximum.

Le barème est donné à titre indicatif entre parenthèses après le numéro de la question.

1 Piles

Exercice 1 (/3) : Donner une implémentation (efficace) des piles en Python et écrire les fonctions de création, d’empilement et de dépilement correspondantes.

Exercice 2 (/3) : Sans disposer de mémoire autre que des piles, combien de piles supplémentaires au moins sont nécessaires pour « retourner » une pile de taille $n \geq 2^1$? Combien d’empilements et de dépilements sont alors effectués ? Écrire un programme en Python pour réaliser cette opération, en utilisant l’implémentation et les fonctions de la question précédente et en supposant qu’il n’y a pas de problème de capacité.

Exercice 3 (/4) : Les piles de cet exercice sont supposées de capacité illimitée et on ne pourra y mettre que des $+$ et des $-$. On représente un entier relatif n sous la forme d’une pile contenant $|n|$ occurrences du symbole marquant le signe de n . Écrire en Python et à l’aide de primitives sur des piles une fonction additionnant deux entiers relatifs, le premier étant représenté comme une pile, et le deuxième étant représenté tel quel. Le résultat doit être représenté comme une pile. Écrire ensuite sous les mêmes conditions une fonction qui additionne deux entiers relatifs représentés tous deux comme des piles. Le résultat doit là aussi être représenté comme une pile.

2 Récursivité

Exercice 4 (/4) : Écrire en Python une fonction récursive `puiss(a,n)` calculant la puissance n -ième d’un nombre a . Déterminer sa complexité en détaillant la formule de récurrence². Écrire ensuite une fonction `puiss_n(n)` qui calcule n^n .

Exercice 5 (/2) : Soit une fonction récursive telle que le coût pour le paramètre $2n$ soit n plus le coût pour le paramètre n , le coût pour le paramètre 1 étant 1. Quelle est la complexité exacte de cette fonction pour le paramètre 2^k , et quel est l’ordre de grandeur pour un paramètre quelconque, si le coût est supposé croissant³ ?

Exercice 6 (/4) : Écrire une fonction récursive appliquant l’algorithme d’Euclide. Écrire ensuite une fonction déterminant un couple de Bézout.

Exercice 7 ([le nombre de points de l’exercice 7]) : Faire l’exercice 7.

1. c’est-à-dire faire en sorte que l’ordre des éléments dans la pile de départ soient inversés

2. bonus si la complexité est logarithmique

3. bonus si un exemple de fonction, même artificiel, est écrit ensuite