

Structures de données et types abstraits

informatique / licence 2

1. On utilise une pile pour effectuer des calculs. Les valeurs sont stockées en haut de la pile et les opérations arithmétiques usuelles « consomment » les deux valeurs supérieures avant d'empiler le résultat de l'opération. Par exemple, pour calculer $2+3$, on empile les valeurs 2 et 3 puis on effectue l'addition, ce que l'on note « 2 3 + » (la pile contient alors la valeur 5). À l'aide de cette notation, indiquer comment s'effectuent les calculs suivants :

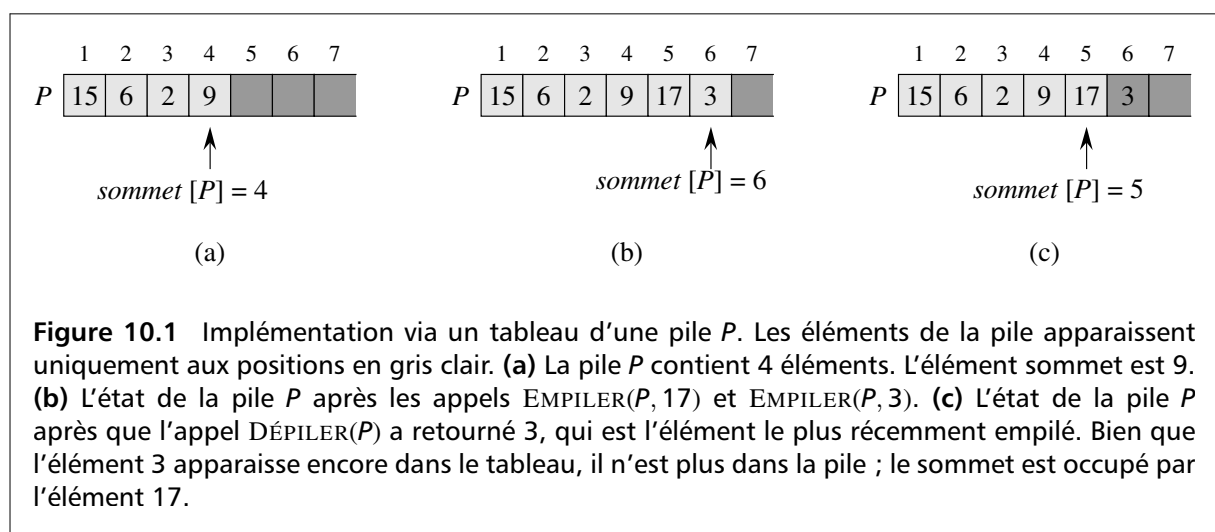
- calcul de 2×3 ;
- calcul de $2+3+4$ (*donner deux versions pour ce calcul*) ;
- calcul de $(2+3) \times (1+4)$;
- calcul de $100-(2+2) \times (4-2)$.

2. Une calculatrice à notation polonaise inversée dispose des instructions suivantes : swap (échange des deux éléments en haut de la pile), rot (récupération du troisième élément en partant du haut de la pile qui passe en haut), dup (duplication de l'élément situé en haut de la pile), add (addition des deux éléments situés en haut de la pile) :

Exemples : 1 2 3 add → 1 5
 1 2 3 swap → 1 3 2
 1 2 3 rot → 2 3 1
 1 2 3 dup → 1 2 3 3

Si l'on part d'une pile contenant les deux valeurs 0 (en bas de la pile) et 1 (en haut de la pile), quelle séquence d'instructions permet d'obtenir à chaque fois un nouveau terme de la suite de Fibonacci ?

3. La figure ci-dessous est extraite de l'ouvrage de référence *Algorithmique* (par Thomas Cormen *et alii*). En s'inspirant de cette figure, illustrer le résultat de chacune des opérations empiler($P, 4$), empiler($P, 1$), empiler($P, 3$), dépiler(P), empiler($P, 8$), dépiler(P) sur une pile P , initialement vide, stockée dans un tableau $P[1..6]$.



Source : *Algorithmique*, Th. Cormen *et alii*, 3ème édition (2010), p. 196.

4. Écrire le pseudo-code des opérations empiler et dépiler pour le système de la question précédente. On admettra ici que les deux opérations sont toujours effectuées dans des situations non problématiques.
5. Améliorer le pseudo-code précédent pour déclencher une erreur le cas échéant (plusieurs causes d'erreur doivent être envisagées ici).
6. On garde l'idée des questions précédentes d'implémenter une pile à l'aide d'un tableau, mais on rend le système plus dynamique en doublant la taille du tableau à chaque fois qu'un débordement risque d'avoir lieu ; il faut alors recopier le contenu de la pile depuis le précédent emplacement de mémoire vers un nouvel emplacement. Montrer que le coût amorti (c'est-à-dire le coût moyen) de l'opération empiler reste un coût constant.
7. Expliquer comment implémenter deux piles dans un seul tableau $A[1..n]$ de telle manière qu'aucune pile ne déborde à moins que le nombre total d'éléments dans les deux piles vaille n . Les opérations empiler et dépiler devront s'exécuter avec un coût constant.
8. Expliquer comment implémenter une pile à l'aide d'une liste simplement chaînée. On attend le pseudo-code des opérations empiler, dépiler et taille (la dernière opération renvoie le nombre d'éléments présents dans la pile).