

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

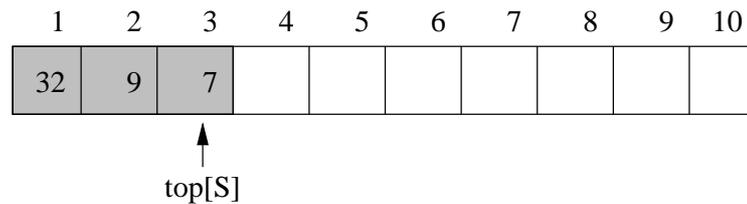
Sections d'Informatique et de Systèmes de Communication

Série d'exercices 3

10 Octobre 2011

1. Stacks

a) Supposons que nous avons un stack implémenté par un array dans l'état initial suivant:



Quel est l'état du stack après les instructions suivantes (c'est-à-dire quelles sont les valeurs dans l'array, et que vaut l'attribut $top[S]$)?

Push($S, 7$), Push($S, Pop(S) + Pop(S)$), Pop(S), Push($S, 11$), Push($S, 12$)

b) Supposons que nous voulons utiliser un stack pour résoudre le problème de vérification de parenthèses pour l'input suivant:

$$[(1 + 2) * 3 - ((4/5) + 6)) * 7]$$

Quel est l'output attendu? En utilisant l'algorithme donné dans le cours avec cet input, quelles sont les opérations effectuées sur le stack, et dans quel ordre? Quel est l'output?

2. Stacks et files d'attente

a) Supposons que nous disposons uniquement de la structure de données stack. Est-il possible de réaliser une file d'attente en utilisant deux stacks? Si oui, donner les algorithmes correspondants pour les opérations de la file d'attente.

b) Nous voulons de nouveau réaliser une file d'attente en utilisant deux stacks, mais avec la condition de plus que le temps de calcul des opérations est similaire à celui d'une implémentation classique de file d'attente.

Plus précisément, supposons que les opérations sur les stacks se font en temps $O(1)$ (i.e. ne dépendent pas de la taille du stack). Donner alors des algorithmes pour les opérations d'une file d'attente basée sur deux stacks qui ont la propriété que, si l'on commence avec une file vide, pour faire les m premières opérations de la file d'attente, on a besoin de $O(m)$ opérations du stack.

3. Permutations avec stacks et files d'attente

Considérons une machine avec les caractéristiques suivantes:

- Elle dispose d'un (seul) stack S .
- Elle peut lire un caractère de l'input et le déposer sur le stack—c'est l'opération **Push**.
- Elle peut enlever un caractère du stack et l'ajouter à l'output—**Pop**.
- Il n'y a pas d'autres opérations ni de stock supplémentaire.

Par exemple, la suite suivante d'instructions

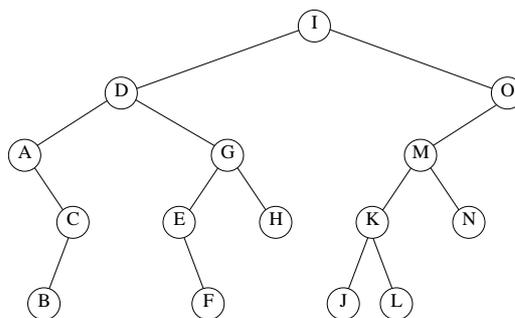
Push, Push, Push, Pop, Push, Pop, Pop, Push, Pop, Pop

créée, si l'input $i = (1, 2, 3, 4, 5)$ est donné, l'output $(3, 4, 2, 5, 1)$.

- Existe-t-il une suite d'instructions qui sort, pour le même input i , l'output $(1, 2, 3, 5, 4)$?
Et l'output $(2, 3, 5, 4, 1)$? Ou encore $(3, 1, 2, 5, 4)$?
- Montrer qu'il est possible d'obtenir la permutation (p_1, p_2, \dots, p_n) de $(1, 2, \dots, n)$ s'il n'existe pas d'indices $i < j < k$ tel que $p_j < p_k < p_i$.
- Si nous remplaçons le stack par une file d'attente, quelles permutations pouvons nous alors obtenir?

4. Arbres

- Donner les suites de sommets obtenues en parcourant l'arbre binaire suivant en l'ordre preorder, inorder et postorder.



- Vérifier qu'il s'agit bien d'un arbre de recherche binaire pour l'ordre alphabétique.
- Prouver qu'un arbre binaire est un arbre de recherche si et seulement si la suite des valeurs obtenue en prenant les clés dans un parcours inorder est croissante.
- Effacer le sommet D de manière à préserver la structure d'arbre de recherche.
- Écrire un algorithme pour effacer dans un arbre de recherche. Donner un argument qui justifie cet algorithme.
- Soit T un arbre binaire de hauteur h . Montrer que T peut avoir au plus $2^{h+1} - 1$ sommets.