

12 novembre 2019

40 minutes.

Tous documents interdits.

Une feuille A4 R/V manuscrite autorisée.

Nom :

Groupe de TD

Prénom :

Exercice (Mélange de liste chaînée)

On cherche à simuler le mélange d'un jeu de 52 cartes selon le principe suivant :

1. on prend un petit tas de cartes sur le dessus du jeu ;
2. on insère ce tas à une position au hasard dans le reste des cartes ;
3. on recommence en 1 autant de fois qu'on veut.

Par exemple, avec seulement 10 cartes et en prenant un petit tas de 3 cartes, puis un petit tas de 5 cartes, on pourrait avoir la transformation suivante :

$$\langle \clubsuit 4, \heartsuit 9, \clubsuit 2, \spadesuit V, \spadesuit 8, \diamond R, \heartsuit D, \diamond 4, \clubsuit 7, \spadesuit D \rangle \implies \langle \spadesuit V, \spadesuit 8, \diamond R, \heartsuit D, \clubsuit 4, \heartsuit 9, \clubsuit 2, \diamond 4, \clubsuit 7, \spadesuit D \rangle$$

$$\langle \spadesuit V, \spadesuit 8, \diamond R, \heartsuit D, \clubsuit 4, \heartsuit 9, \clubsuit 2, \diamond 4, \clubsuit 7, \spadesuit D \rangle \implies \langle \heartsuit 9, \clubsuit 2, \diamond 4, \spadesuit V, \spadesuit 8, \diamond R, \heartsuit D, \clubsuit 4, \clubsuit 7, \spadesuit D \rangle$$

On suppose ici que le jeu de cartes est représenté par une séquence sous forme de liste chaînée, chaque cellule représentant une carte. Proposer une solution algorithmique et son analyse pour ce problème.

Recommandations

- Donnez les définitions des structures de données qui seront nécessaires.
- Montrez sur un exemple (*différent de ceux ci-dessus*) le comportement de l'algorithme *en dessinant les détails bas-niveau des listes chaînées*.
- Donnez l'algorithme en pseudo-code. **Attention** : il est conseillé de donner l'algorithme général et de séparer les opérations bas-niveau dans des fonctions auxiliaires.
- Faites une analyse de complexité de votre algorithme.
- Pourrait-on améliorer l'efficacité de votre algorithme ? Si oui, expliquez rapidement les modifications nécessaires. Si non, expliquez en quoi les choix algorithmiques que vous avez fait le rendent efficace.

Pour modeliser les cartes, j'utilise la structure de donnees suivante :

```
type Carte {
  valeur : entier
  couleur : caractere
  suivant : reference vers Carte
}
```

Les valeurs sont entre 1 et 13, l'As represente par 1 et valet, dame, roi par 11, 12, 13.
Les couleurs sont 'C', 'P', 'T' et 'K' pour differencier carreau de coeur.

Un tas sera represente ainsi :

```
type Tas {
  tete : reference vers Carte
}
```

(note : on pourrait egalement avoir simplement un champ "valeur" et une convention du type 101 a 113 pour coeurs, 201 a 213 pour piques, etc.)

Le melange se fera alors ainsi :

Melange (T) :

n <- longueur (T)

iterations <- alea (400, 600)

repete iterations fois :

un_melange (T, n)

 $O(1)$
 $O(n)$

(je considere comme existante une fonction alea (a, b) qui renvoie un nombre au hasard entre a et b inclus en $O(1)$)

complexité' = $O(i \times n)$
avec i le nombre de melanges et n nombre de cartes

Longueur se calcule aisement ainsi :

longueur (T) :

n = 0

cel <- T.tete

tant que cel != Nil :

n <- n+1

cel <- cel.suivant

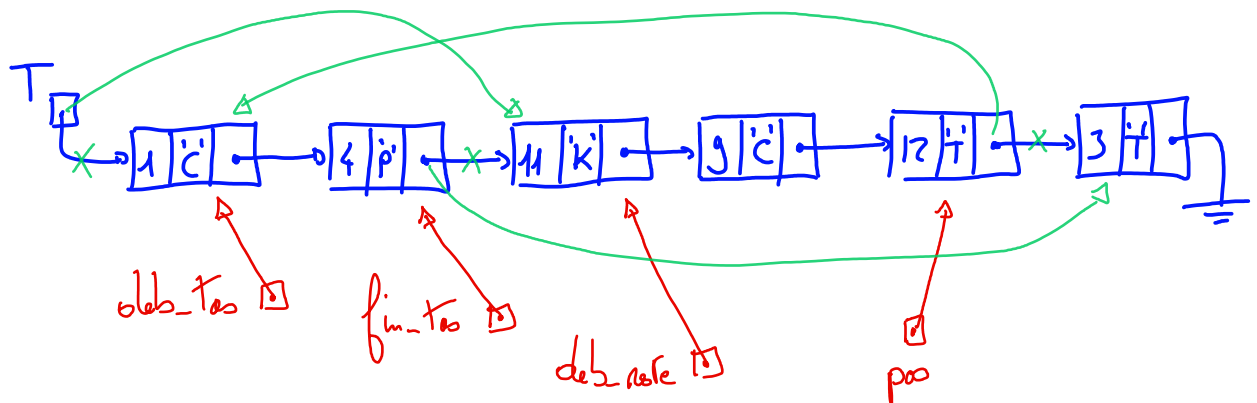
retourner n

$O(1)$
longueur $\times O(1)$ } $O(n)$ avec n le nombre de cartes

Pour "un_melange", je prends deux nombres au hasard : "taille" entre 5 et 15, puis "position" entre 1 et n-taille avec n le nombre de cartes. On "coupera" ensuite la liste chainee apres "taille" cellules, et on ajoute le morceau coupe apres "position" cellules dans ce qu'il reste.

J'utilise des pointeurs pour reperer les cellules importantes qui necessiteront des changements de liens de chainage : deb_tas, fin_tas, deb_reste, et pos.

Voici un exemple avec 6 cartes, taille = 2 et position = 3. En vert les modifications de chainage a effectuer.



un_melange (T, n) :

assert (n > 15)

taille <- alea (5, 15)

position <- alea (1, n-taille)

deb_tas <- T.tete

fin_tas <- avance (deb_tas, taille-1)

deb_reste <- fin_tas.suivant

pos <- avance (deb_reste, position-1)

T.tete <- deb_reste

fin_tas.suivant <- pos.suivant

pos.suivant <- deb_tas

$O(1)$

$O(1)$
 $O(taille)$
 $O(1)$
 $O(n-taille)$

$O(1)$

$O(x)$

avance (cel, x) :
 repete x fois :
 cel <- cel.suivant
 retourner cel

(note : on a ici une
 modification locale de cel, la
 variable n'est pas modifiee
 dans l'appelant)

$O(1) + O(taille) + O(n-taille)$

$= O(n)$

La complexite pour un melange ne peut pas etre amelioree avec cette structure de donnees.

Dans le pire des cas, la position choisie aleatoirement se trouve en derniere position et il faut donc parcourir toute la liste chainee pour la trouver.

En termes d'efficacite generale, on n'a egalement qu'au plus un seul parcours de liste dans "un_melange" grace a l'utilisation de variables qui pointent vers les cellules importantes. Par exemple, cela evite de reparcourir le petit tas pour chercher la queue quand on cherche a l'insérer.

12 novembre 2019

40 minutes.

Tous documents interdits.

Une feuille A4 R/V manuscrite autorisée.

Nom :

Groupe de TD

Prénom :

Exercice (Mélange de liste chaînée)

On cherche à simuler le mélange d'un jeu de 52 cartes selon le principe suivant :

1. on prend un petit tas de cartes au hasard quelque part dans le jeu ;
2. on repose ce tas au dessus du reste des cartes ;
3. on recommence en 1 autant de fois qu'on veut.

Par exemple, avec seulement 10 cartes et en prenant un petit tas de 3 cartes, puis un petit tas de 5 cartes, on pourrait avoir la transformation suivante :

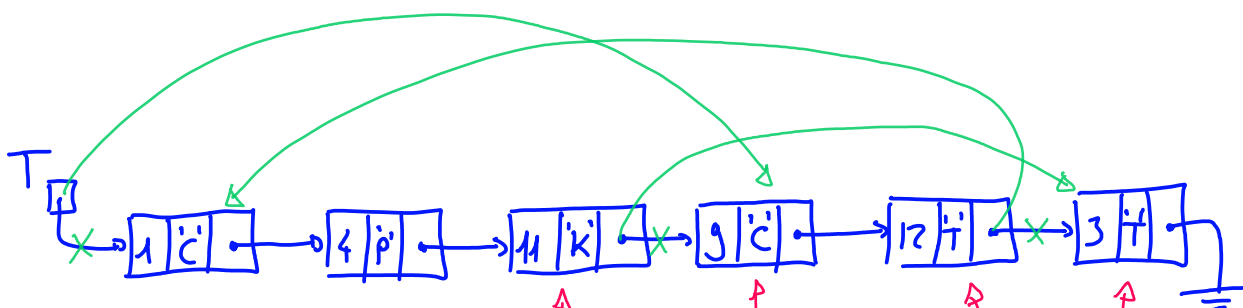
$$\begin{aligned} \langle \clubsuit 4, \heartsuit 9, \clubsuit 2, \spadesuit V, \spadesuit 8, \diamond R, \heartsuit D, \diamond 4, \clubsuit 7, \spadesuit D \rangle &\Rightarrow \langle \spadesuit 8, \diamond R, \heartsuit D, \clubsuit 4, \heartsuit 9, \clubsuit 2, \spadesuit V, \diamond 4, \clubsuit 7, \spadesuit D \rangle \\ \langle \spadesuit 8, \diamond R, \heartsuit D, \clubsuit 4, \heartsuit 9, \clubsuit 2, \spadesuit V, \diamond 4, \clubsuit 7, \spadesuit D \rangle &\Rightarrow \langle \heartsuit D, \clubsuit 4, \heartsuit 9, \clubsuit 2, \spadesuit V, \spadesuit 8, \diamond R, \diamond 4, \clubsuit 7, \spadesuit D \rangle \end{aligned}$$

On suppose ici que le jeu de cartes est représenté par une séquence sous forme de liste chaînée, chaque cellule représentant une carte. Proposer une solution algorithmique et son analyse pour ce problème.

Recommandations

- Donnez les définitions des structures de données qui seront nécessaires.
- Montrez sur un exemple (*différent de ceux ci-dessus*) le comportement de l'algorithme *en dessinant les détails bas-niveau des listes chaînées*.
- Donnez l'algorithme en pseudo-code. **Attention** : il est conseillé de donner l'algorithme général et de séparer les opérations bas-niveau dans des fonctions auxiliaires.
- Faites une analyse de complexité de votre algorithme.
- Pourrait-on améliorer l'efficacité de votre algorithme ? Si oui, expliquez rapidement les modifications nécessaires. Si non, expliquez en quoi les choix algorithmiques que vous avez fait le rendent efficace.

L'exercice est très similaire au précédent et nous ne reprenons ici que ce qui diffère.



un_melange (T, n) :

...

```
pos <- avance (T.tete, position-1)
deb_tas <- pos.suivant
fin_tas <- avance (deb_tas, taille - 1)
deb_reste <- fin_tas.suivant
```

```
fin_tas.suivant <- T.tete
pos.suivant <- deb_reste
T.tete <- deb_tas
```