14 novembre 2018
40 minutes.
Tous documents interdits.
Une feuille A4 R/V manuscrite autorisée.

Nom : .....

Prénom : .....

Groupe de TD

Exercice 1 (Permutations d'entiers)

Dans cet exercice, on s'intéresse à des permutations d'entiers. On veut stocker tous les entiers de 1 à n dans un ordre aléatoire dans une séquence S sous forme de **listes chaînées**.
On a l'algorithme haut-niveau ci dessous à gauche, qui insère des cellules à des positions aléatoires. On s'intéresse à l'implantation bas-niveau ci-dessous à droite de la fonction `insere_position`.

```
S ← nouvelle Séquence
pour i de 1 à n faire
    p ← aléatoire(0, i − 1)
    insere_position(S, p, i)
retourner S
```

```
insere_position(S, p, val)
    ncel ← nouvelle Cellule
    ncel.valeur ← val
    A:
    cel ← S.tête
    i ← 0
    pour i de 1 à p − 1 faire
        B:
    ncel.suivant ← C:
    cel.suivant ← D:
```

Consigne : compléter les trous de l'algorithme :

A : (cas particulier)

B :

C :

D :

Exercice 2 (Analyse d'algorithme)

Dans cet exercice, on part de la séquence S aléatoire de l'exercice précédent. On suppose donc que tous les entiers de 1 à n sont stockés dans un ordre aléatoire dans S.

On considère l'algorithme décrit informellement ci-dessous :

1. on commence par chercher l'entier 1 dans S ;
2. on le déplace à la **fin** de S ;
3. on cherche l'entier 2, que l'on déplace au **début** de S ;
4. on recommence avec l'entier 3, et ainsi de suite jusqu'à n (les impairs à la fin, les pairs au début).

Consignes Vous devez procéder uniquement par **modification des liens de chaînage**.

1. Commencez par montrer sur un ou deux exemples de déplacement le comportement de l'algorithme (dessinez la liste chaînée et les modifications effectuées).
2. Donnez l'algorithme en pseudo-code. **Attention** : il est demandé de donner l'algorithme en haut-niveau et de détailler les opérations bas-niveau dans des fonctions séparées.
3. Faites une analyse de complexité de votre algorithme.
4. *Bonus :* commentez rapidement l'effet de cet algorithme sur la séquence.

14 novembre 2018
40 minutes.
Tous documents interdits.
Une feuille A4 R/V manuscrite autorisée.

Nom : .....

Prénom : .....

Groupe de TD

Exercice 1 (Permutations d'entiers)

Dans cet exercice, on s'intéresse à des permutations d'entiers. On veut stocker tous les entiers de 1 à n dans un ordre aléatoire dans une séquence S sous forme de **listes chaînées**.
On a l'algorithme haut-niveau ci dessous à gauche, qui insère des cellules à des positions aléatoires. On s'intéresse à l'implantation bas-niveau ci-dessous à droite de la fonction `insere_position`.

```
S ← nouvelle Séquence
pour i de 1 à n faire
    p ← aléatoire(0, i − 1)
    insere_position(S, p, i)
retourner S
```

```
insere_position(S, p, val)
    ncel ← nouvelle Cellule
    ncel.valeur ← val
    A:
    cel ← S.tête
    i ← 0
    pour i de 1 à p − 1 faire
        B:
    ncel.suivant ← C:
    cel.suivant ← D:
```

Consigne : compléter les trous de l'algorithme :

A : (cas particulier)

B :

C :

D :

Exercice 2 (Analyse d'algorithme)

Dans cet exercice, on part de la séquence S aléatoire de l'exercice précédent. On suppose donc que tous les entiers de 1 à n sont stockés dans un ordre aléatoire dans S.

On considère l'algorithme décrit informellement ci-dessous :

1. on commence par chercher l'entier 1 dans S ;
2. on le décale de 1 position plus loin dans S ;
3. on recommence avec l'entier 2 que l'on décale de 2 positions ;
4. on procède de même jusqu'à n ; si le décalage fait « sortir » l'entier de la séquence, on le place en queue de la séquence.

Consignes Vous devez procéder uniquement par **modification des liens de chaînage**.

1. Commencez par montrer sur un ou deux exemples de déplacement le comportement de l'algorithme (dessinez la liste chaînée et les modifications effectuées).
2. Donnez l'algorithme en pseudo-code. **Attention** : il est demandé de donner l'algorithme en haut-niveau et de détailler les opérations bas-niveau dans des fonctions séparées.
3. Faites une analyse de complexité de votre algorithme.
4. *Bonus :* commentez rapidement l'effet de cet algorithme sur la séquence.