Feuille d'exercices n°5 - Structures

Notions abordées

- évaluation paresseuse des opérateurs `&&` et `||`
- révisions sur les notations de Landau
- étude de la complexité amortie de la table dynamique
- structure pour un multi-ensemble

Structures séquentielles

Exercice 1 Consultations dans un cabinet sans rendez-vous

Dans cet exercice on cherche à modéliser les consultations dans un cabinet sans prise de rendez-vous. Ainsi lorsqu'une personne arrive pour une consultation, elle est placée en salle d'attente et sera appelée plus tard. On ne sait pas quand elle sera appelée, mais elle sera appelée avant les personnes arrivées après elle.

Question 1 ✍

Quelle structure de données abstraite pourrait modéliser cette situation ? Préciser quel serait le type des éléments de cette structure, puis proposer une représentation schématique de cette structure.

Question 2 ✍

Dans le cas où les heures supplémentaires sont autorisées, on arrête la liste des consultations du jour une heure avant la fermeture. Comment produire la liste de toutes les consultations restant à réaliser ce jour là ?

Question 3 ✍

Dans le cas où les heures supplémentaires ne sont pas autorisées, on peut être amener à limiter le nombre de consultations durant le reste du temps imparti. Envisageons le cas où, une heure avant la fermeture du cabinet, on annonce aux dix prochaines personnes qu'elles seront reçues le jour même, puis on annonce aux autres personnes qu'elles ne seront pas appelées ce jour là. Comment réaliser ces annonces ?

Question 4 ✍

Le plus réaliste est sûrement qu'on ait un peu de souplesse, mais des limites réelles quand même. On suppose qu'une heure avant la fermeture on arrête la liste des prochaines consultations aux 10 prochaines, auxquelles on ajoute celles des personnes qui attendent depuis déjà 20 minutes, dans la limite de 15 consultations en tout. Comment réaliser ces annonces ?

Question 5 ✍

La structure de données permet-elle la gestion de désistements ?

Exercice 2 Gestions d'espaces de travail

On souhaite modéliser la gestion d'espaces de travail à la manière de l'environnement `GNOME`.
Un espace de travail peut être vide, ou bien contenir des fenêtres. Les espaces de travail sont numérotés, à partir de 1. Il y a toujours au moins un espace de travail, et toujours au moins un espace vide, car le dernier espace est toujours vide.
On est toujours sur un des espaces de travail. On suppose qu'on ne peut changer d'espace de travail que pour aller vers celui de juste après, ou celui de juste avant. Pour ouvrir une fenêtre dans un espace de travail, il faut être positionné sur cet espace.
Au maximum il y a deux espaces de travail vides, car lorsqu'on quitte un espace de travail vide il est supprimé, à moins d'être le dernier.

Dans la suite de l'exercice on suppose qu'on a déjà une structure de données modélisant les fenêtres.

Question 1
Identifier les informations qu'un espace de travail doit contenir intrinsèquement, *i.e.* indépendamment de comment il est articulé avec les autres espaces potentiels.

Question 2
Identifier les opérations que l'on doit pouvoir réaliser.

Question 3
Quelle structure de données permettrait de modéliser l'ensemble des espaces de travail ? Expliquer comment les opérations listées à la question précédente serait réalisée, et préciser quelle serait leur complexité.

Question 4
Si l'on impose que les numéros doivent être successifs, comment fait-on ?

Question 5
Peut-on garder une complexité constante pour l'opération de fermeture d'une fenêtre en imposant que les numéros *semblent* successifs sous l'hypothèse que l'on n'a pas de visualisation globale de tous les espaces de travail avec leur numéro, mais seulement le numéro de l'espace courant et le nombre total d'espaces ?

Exercice 3 Petite étude graphique de $x \mapsto 2^{\lfloor \log_2(x) \rfloor}$

Question 1
Sur une feuille à petits carreaux, tracer le graphe des fonctions $f : x \mapsto \lfloor \log_2(x) \rfloor$ et $g : x \mapsto 2^{\lfloor \log_2(x) \rfloor}$ sur l'intervalle $[1, 33]$.

Question 2
Repérer sur le graphique les valeurs de x pour lesquelles $g(x) = x$.

Question 3
Comparer graphiquement la fonction $id = x \mapsto x$ et g. Que pouvez-vous en déduire ?

Question 4
En s'appuyant sur le graphique, proposer un équivalent de $2^{\lfloor \log_2(n) \rfloor}$, puis le démontrer.

Remarque : dans la même veine on pourrait donner un équivalent de $\lfloor \sqrt{n} \rfloor^2$.

Exercice 4 Complexité amortie d'ajouts successifs dans une table dynamique

On s'intéresse dans cet exercice à la structure de table dynamique. L'ajout d'un élément dans cette structure se fait en temps constant si la structure n'est pas pleine, et en temps linéaire si elle l'est, car on alloue un espace mémoire deux fois plus grande, et on y recopie tous les éléments. On s'intéresse au **coût amorti** de l'opération d'ajout, c'est-à-dire au coût moyen d'une opération d'ajout lorsqu'on réalise des ajouts successivement sur la même structure, (à ne pas confondre avec une complexité moyenne).
Plus précisément, étudions le coût de $N \in \mathbb{N}^*$ opérations d'ajout successives sur une table dynamique initialement de taille 1. Pour $j \in [1..N]$, on note σ_j le coût de le j-ième opération d'ajout, et on note $S_N = \sum_{j=1}^{N} \sigma_j$ le coût cumulé des N opérations.

Question 1 ✍
Pour $j \in [1..N]$, donner la valeur de σ_j.

Question 2 ✍
On note D la plus grande puissance de 2 inférieure ou égale à $N-1$ et P son exposant. Donner explicitement les valeurs de P et D.

Question 3 ✍
Calculer précisément S_N en fonction de N, P et D.

Question 4 ✍
Soient u, v, w trois suites de $(\mathbb{R}^+)^{\mathbb{N}}$. Si $u \in \Theta(v)$ et $w \in O(v)$, a-t-on $u + w \in \Theta(v)$?
Le démontrer ou fournir un contre-exemple

Question 5 ✍
En déduire un équivalent de S_N, et conclure sur le coût amorti d'une opération d'ajout dans une table dynamique.

Exemple de structure non séquentielle

Exercice 5 Multi-ensemble

Un **multi-ensemble** est un objet mathématique représentant une collection non ordonnée d'éléments. À la différence d'un ensemble, un multi-ensemble peut contenir plusieurs occurrences d'un même élément. On appelle alors **multiplicité** d'un élément son nombre d'occurrences. Un multi-ensemble M d'éléments d'un ensemble S peut-être décrit par une fonction de S dans $\mathbb{N}$ associant à chaque élément sa multiplicité.

On notera les multi-ensembles entre doubles accolades, par exemple $X = \{\{\pi, \sqrt{2}, \pi, \sqrt{2}, e, \sqrt{2}, e, 1.5\}\}$ est le multi-ensemble qui contient deux fois l'élément e, trois fois l'élément $\sqrt{2}$, deux fois l'élément π et une fois l'élément 1.5. On note que $X = \{\{\pi, \pi, \sqrt{2}, \sqrt{2}, \sqrt{2}, e, e, 1.5\}\}$.

Question 1 ✍

Donner la représentation de X sous forme de fonction.

On appelle **cardinal** d'un multi-ensemble son nombre d'éléments comptés avec multiplicité. Si x apparaît n fois dans le multi-ensemble, il compte pour n dans le cardinal du multi-ensemble.

Question 2 ✍

Donner le cardinal de X.

Dans toute la suite on s'intéresse à des multi-ensembles de cardinal fini, d'éléments de type quelconque `type_elem`. On se propose une représentation en C des multi-ensembles permettant de faire les opérations ensemblistes usuelles d'ajout d'un élément et d'union.

Question 3 ✍

Définir en C les structures nécessaires à la représentation de multi-ensemble.

Question 4 ✍

Donner donner une représentation schématique de la mémoire représentant le multi-ensemble suivant : $\{\{1, 2, 5, 3, 5, 1, 2, 5\}\}$.

Question 5 ✍

Donner une fonction `multiset empty();` ne prenant aucun argument et retournant un objet représentant le multi-ensemble vide.

Question 6 ✍

Donner une fonction `cardinal` retournant le cardinal d'un multi-ensemble.

Question 7 ✍

Expliquer comment, avec les structures de données proposées, gérer l'ajout d'un élément, avec multiplicité, à un multi-ensemble.