2I003 – Interrogation longue 1 (1h)

19 octobre 2017

L'interrogation est notée sur 25 points. Elle consiste en un problème composé de deux parties. La deuxième partie dépend de la première, mais **chaque question peut être traitée en admettant les questions précédentes**.

Le but du problème est d'étudier un algorithme qui calcule la signature d'une permutation.

1 Première partie (11pts).

Soient deux entiers $0 \leq j < n$ et `T` un tableau de taille n contenant des entiers tous distincts. Dans cette partie préliminaire, on étudie une fonction `nbSup` qui, étant donnés `T` et j, doit retourner le nombre d'incides $i \in \{0, \ldots, j-1\}$ tels que `T[i] > T[j]`.

La fonction `nbSup` est définie comme suit :

```
def nbSup(T,j):
    res = 0
    i = 0
    while i < j:
        if T[i] > T[j]:
            res = res + 1
        i = i + 1
    return res
```

Question 1 (2pts). Montrer que la fonction `nbSup(T,j)` se termine pour tout tableau `T` de taille n et tout $j \in \{0, \ldots, n-1\}$.

Pour $i \in \{0, \ldots, j\}$, on définit r_i comme la valeur de `res` à la fin du i-ème tour de boucle (c'est-à-dire, après que l'instruction `i = i + 1` ait été effectuée pour la i-ème fois). Par convention, $r_0 = 0$. On définit aussi la propriété

$$\Pi(i) : \text{« } r_i \text{ est le nombre d'éléments } k \in \{0, \ldots, i-1\} \text{ tels que } \texttt{T[k] > T[j]} \text{ ».}$$

Question 2 (4pts). Montrer que pour tout $i \in \{0, \ldots, j\}$, la propriété $\Pi(i)$ est vérifiée.

Question 3 (2pts). En déduire la validité de la fonction `nbSup`.

Question 4 (3pts). Exprimer en fonction de j la complexité exacte de `nbSup(T,j)`, en terme de nombre de comparaisons entre éléments du tableau.

2 Seconde partie (14pts).

Question 5 (1pt). Donner sans justifier la valeur de $(-1)^u$ suivant si u est pair ou impair.

On note S_n l'ensemble des permutations de $\{1,\dots,n\}$. Une permutation $\sigma \in S_n$ peut être représentée en machine par un tableau `T` tel que `T[i - 1]` $= \sigma(i)$ pour tout $i \in \{1,\dots,n\}$ (le décalage d'indice est dû au fait que le premier indice d'un tableau est 0).

Pour toute permutation $\sigma \in S_n$, on définit la **signature** de σ comme

$$\varepsilon(\sigma) := \prod_{1\le i<j\le n} \text{signe}(\sigma(j)-\sigma(i))\,,$$

où la fonction signe est définie par $\text{signe}(x) := \begin{cases} -1 & \text{si } x<0\,,\\ 1 & \text{si } x\ge 0\,.\end{cases}$

On **admet** que l'on a $\varepsilon(\sigma) = \prod_{j=2}^{n}(-1)^{u_j}$ où u_j est le nombre d'entiers $i \in \{1,\dots,j-1\}$ tels que $\sigma(i) > \sigma(j)$.

Soit enfin la fonction `signature(T,n)`, qui a pour but de retourner la signature d'une permutation $\sigma \in S_n$, qui encodée comme un tableau `T` de taille n.

```
def signature(T,n):
    if (n == 1):
        res = 1
    else:
        res = signature(T,n-1)
        u = nbSup(T,n-1)
        if (u % 2 == 1):
            res = - res
    print("u = ", u, "et signature(", T, ", ", n, ") = ", res)
    return res
```

Question 6 (4pts). Exécuter `signature(T,n)` avec `T = [3,2,4,1]` et `n` $= 4$, puis donner sa valeur de retour.

Question 7 (6pts). Démontrer la terminaison et la validité de la fonction `signature(T,n)`. On pourra montrer la propriété $\mathcal{P}(k)$: « `signature(T,`k`)` termine et renvoie $\prod_{j=2}^{k}(-1)^{u_j}$ » pour des valeurs appropriées de k.

Question 8 (3pts). Exprimer en fonction de n la complexité de `signature(T,n)` en nombre de comparaisons entre éléments du tableau `T` de taille n.