TP n°9 - Recherche du k-ième plus grand élément

Avant de commencer

- Récupérer sur le site de classe les fichiers `tp09_fichier_reponses.txt` et `tp09_fichier_reponses_exemple.txt`
- Noter dans le fichier `tp09_fichier_reponses.txt` votre valeur pour u_0 indiquée dans le fichier `tp09_u0.txt` présent sur le site de classe.
- Pour les questions commençant par la mention « **Fichier réponses** » :
  - Pour ceux qui sont au lycée, compléter la partie correspondante dans le fichier `tp09_fichiers_reponses.txt` ;
  - Pour ceux qui sont à distance, répondre à la question correspondante dans le questionnaire envoyé par mail.
- *Remarques :*
  - Le fichier `tp09_fichier_reponses_exemple.txt` contient les réponses pour une autre valeur de u_0 (qui n'est bien évidemment pas celle que vous devez utiliser pour remplir votre fichier réponses).
  - Je vous conseille de définir au début de votre script deux variables correspondant respectivement à votre valeur de u_0 et à celle du fichier exemple pour éviter d'éventuelles fautes de frappe lors de vos tests et calculs.

Une entreprise de n employés ($n \geqslant 2$) veut récompenser k ses meilleurs vendeurs. Chaque employé, désigné par son numéro de badge i (où $0 \leqslant i < n$), a réalisé un chiffre de ventes a_i au cours de l'année 2020. On suppose que pour tout entier i, a_i est un entier. L'objectif est de déterminer le k-ième chiffre de ventes le plus élevé, ce qui correspondra au seuil à partir duquel les employés auront le droit à une prime.

Dans tout le sujet, on pourra utiliser `Array.make` et `Array.length`, mais pas d'autres fonctions du module `Array`

Préliminaires

On considère une suite d'entiers $(u_k)_{k\in\mathbb{N}}$ vérifiant la relation de récurrence :

$$\forall k \in \mathbb{N}, u_{k+1} = (15091 \times u_k) \bmod 64007$$

1. Écrire une fonction

   ```
   suivant : int -> int
   ```

   qui, lorsqu'elle prend en argument l'entier correspondant à u_k, renvoie u_{k+1}.
2. Écrire une fonction

   ```
   tab_u : int -> int -> int array
   ```

   qui prend en argument la valeur de u_0 et un entier n et qui renvoie le tableau $[|u_0; u_1; \ldots; u_{n-1}|]$ des n premières valeurs de la suite u.
3. **Fichier réponses :** Donner les valeurs de u_{20}, u_{2000}, u_{20000}.

Dans toute la suite, on considère que le tableau des n chiffres de ventes (n pouvant dépendre de la question) est donné par `tab_u u0 n` où `u0` est votre valeur pour u_0.

Partie I

4. Dans cette question, on suppose que $k = 1$.
   (a) Écrire une fonction

   ```
   meilleure_vente : 'a array -> 'a
   ```

   qui prend en argument le tableau des chiffres de ventes et qui renvoie le meilleur chiffre de ventes.
   (b) **Fichier réponses :** Donner la valeur de la meilleure vente pour $n = 1000$.
5. Dans cette question, on suppose que la distribution des chiffres de ventes est relativement uniforme. Une approximation du seuil de ventes à atteindre pour obtenir la prime est alors donnée par la moyenne du chiffre de ventes minimum et du chiffre de ventes maximum, respectivement pondérés par $(k-1)$ et $n-k$.
   (a) Écrire une fonction

   ```
   approx_seuil: int array -> int -> int
   ```

   prenant en argument le tableau des chiffres de ventes et l'entier k et renvoyant cette moyenne pondérée arrondie à l'entier inférieur.
   (b) **Fichier réponses :** Indiquer le seuil obtenu avec la fonction précédente pour $k = 200$ et $n = 5000$.

Partie II

On traite maintenant le cas général, en ne faisant aucune hypothèse sur la répartition des ventes.
Pour calculer le k-ième meilleur chiffre de ventes, on va trier partiellement le tableau dans l'ordre décroissant, de sorte que les k plus grands éléments soient à leur place dans le tableau.

6. Écrire une fonction

```
echange : 'a array -> int -> int -> unit
```

qui prend en argument un tableau et deux entiers i et j et échangeant en place les éléments d'indices i et j du tableau. Aucune vérification sur i et j n'est demandée.

7. Écrire une fonction

```
seuil : 'a array -> int -> 'a
```

prenant en argument le tableau des chiffres de ventes et l'entier k et renvoyant le k-ième plus grand élément du tableau.

8. **Fichier réponses :** Indiquer le seuil obtenu avec la fonction précédente pour $k = 200$ et $n = 5000$.

Partie III

Le nombre d'employés n étant particulièrement grand, on cherche à optimiser le calcul du k-ième meilleur chiffre de ventes.

Pour cela, on procède comme suit : on choisit un employé i, et on place son chiffre de ventes $v = a_i$ à l'indice l qu'il aurait dans le tableau si celui-ci était trié dans l'ordre décroissant, en réordonnant le tableau de sorte que $a_j > v$ si $j < l$ et $a_j \leqslant v$ et $j > l$. Si l'indice l correspond au k-ième élément du tableau, on a terminé, sinon, on recommence avec la partie à gauche ou à droite de l dans le tableau.

9. Soient g et d deux entiers tels que $0 \leqslant g < d < n$ et soit $v = a_g$. Écrire une fonction

```
partition : 'a array -> int -> int -> int
```

qui prend en argument le tableau des chiffres de ventes et les entiers g et d (dans cet ordre), qui réordonne le tableau en place entre les indices g et d compris de sorte que $a_j > v$ pour $g \leqslant j < l$ et $a_j \leqslant v$ pour $l < j \leqslant d$, et qui renvoie l'indice l de la valeur v après modification du tableau.
Indication : On procédera par des échanges ; on pourra laisser v à l'indice g jusqu'à avoir fini de parcourir la portion de tableau puis procéder à un dernier échange avec l'indice l.

10. **Fichier réponses :** Indiquer l'indice obtenu avec la fonction précédente pour $n = 5000$, $g = 1000$ et $d = 2000$.

11. Écrire une fonction récursive

```
seuil_rec : 'a array -> int -> int -> int -> 'a
```

qui prend en argument le tableau des chiffres de ventes, trois entiers j, g et d (dans cet ordre) tels que $0 \leqslant g \leqslant j \leqslant d$ et qui renvoie l'élément qui serait à l'indice j dans le tableau des ventes trié dans l'ordre décroissant. On pourra utiliser la fonction `partition`.

12. En déduire une fonction

```
seuil_bis : 'a array -> int -> 'a
```

prenant en argument le tableau des chiffres de ventes et l'entier k et renvoyant le k-ième plus grand élément du tableau.

13. **Fichier réponses :** Indiquer le seuil obtenu avec la fonction précédente :
    (a) pour $k = 1000$ et $n = 20000$;
    (b) pour $k = 10000$ et $n = 500000$.