

Activité – Écriture des algorithmes

I. Algorithme de recherche d'un élément

Activité d'introduction : un tableau, des feuilles et des aimants

On dispose de 6 feuilles de papiers. Sur chacune d'elle une valeur entière a été écrite. Ces 6 feuilles de papier sont aimantées au tableau, faces cachées. On ne peut voir la face visible que d'une seule carte à la fois. On dispose au tableau d'une case mémoire appelée `trouvee` dans laquelle on peut stocker une valeur booléenne.

1. Décrire la démarche permettant de répondre à la question : l'entier v est-il écrit sur l'une des 6 feuilles ?

2. Dans quel cas faut-il absolument retourner toutes les feuilles de papier pour répondre ?

Spécification de l'algorithme

La spécification d'un algorithme précise de manière non ambiguë ce que doit faire un algorithme. En particulier, on y indique : le nom des données manipulées (entrées), le nom des données renvoyées (sorties), le rôle de l'algorithme ainsi que les hypothèses sur les entrées (précondition) et les sorties (postcondition). Un algorithme sera considéré comme correct s'il respecte cette spécification.

Voici la spécification de l'algorithme de recherche d'un élément dans un tableau :

- Entrées : tableau T de taille n , une valeur v
- Sortie : un booléen `trouve`
- Rôle : chercher si la valeur v est présente dans T
- Précondition : v est du même type que les éléments de T
- Postcondition : `trouve` vaut VRAI si $v \in T$ et FAUX sinon

Algorithme

Idée de l'algorithme :	
------------------------	--

Coût de l'algorithme

3. Dans le pire cas, combien de passages y a-t-il dans la boucle Pour ?

4. Dans le pire cas, combien de fois la condition $T[i] = v$ est-elle évaluée ?

5. Dans le pire cas, combien de fois l'affectation `trouvee ← Vrai` est-elle effectuée ?

6. Complétez tableau suivant donnant le nombre d'opérations élémentaires dans le pire cas :

Nombre d'affectations	Nombre de comparaisons	Nombre d'opérations arithmétiques

Définition

Le **coût d'un algorithme** est le nombre d'opérations élémentaires (arithmétiques ou logiques) ainsi que d'affectations nécessaires à son exécution **dans le pire cas**.

Le coût d'un algorithme dépend toujours de la taille des données d'entrée (ici la taille n du tableau).

7. Ecrivez une fonction Python appelée `appartient` qui prend en paramètres une valeur v et un tableau T et renvoie `True` si la valeur v est présente dans le tableau T et `False` sinon.

II. Algorithme de recherche du maximum

Activité d'introduction : un tableau, des feuilles et des aimants

On dispose de 6 feuilles de papiers. Sur chacune d'elle une valeur entière a été écrite. Ces 6 feuilles de papier sont aimantées au tableau, faces cachées. On ne peut voir la face visible que d'une seule carte à la fois. On dispose au tableau d'une case appelée `maxi` dans laquelle on peut stocker une valeur (entière).

1. Décrire la démarche permettant de répondre à la question : quelle est la valeur maximale ?

2. Dans quel cas faut-il absolument retourner toutes les feuilles de papier pour répondre ?

Spécification de l'algorithme

- Entrées : tableau T de taille n
- Sortie : entier max_i
- Rôle : chercher l'élément maximal de T
- Précondition : T est un tableau non vide d'entiers
- Postcondition : max_i est un entier, max_i est l'élément maximal de T

Algorithme

Coût de l'algorithme

3. Donnez un exemple de tableau que l'on peut considérer comme le pire cas en termes d'opérations élémentaires.
4. Complétez tableau suivant donnant le nombre d'opérations élémentaires dans le pire cas :

Nombre d'affectations	Nombre de comparaisons	Nombre d'opérations arithmétiques

5. Quel est le coût de cet algorithme ?
6. Ecrire une fonction Python appelée `maximum` qui prend en paramètres un tableau T d'entiers et renvoie la valeur maximale du tableau T .

III. Algorithme de calcul d'une moyenne

On souhaite écrire un algorithme qui calcule la moyenne des valeurs d'un tableau T d'entiers.

Spécification de l'algorithme

1. Proposez une spécification d'un tel algorithme.

- Entrées :
- Sortie :
- Rôle :
- Précondition :
- Postcondition :

Algorithme

2. Proposez un algorithme correspondant à cette spécification.

3. Quel est le coût de cet algorithme ?

Nombre d'affectations	Nombre de comparaisons	Nombre d'opérations arithmétiques

4. Ecrivez une fonction Python appelée `moyenne` qui prend en paramètres un tableau `T` d'entiers, non vide, et renvoie la moyenne des valeurs du tableau `T`. On veut une version avec un parcours par indice et une autre avec un parcours par valeur.

Références

- Cours du DIU EIL, Université de Nantes, C. JERMANN.
- C. IZAC, G. OLLIVIER, pour l'idée de la première activité d'introduction lors du DIU EIL (Nantes).