

Le parcours séquentiel

Christian LAU, Raihei BOOSIE, Tautu TAMATAI

1. Principe

Une **séquence** est une suite d'éléments de même type : tableau d'entiers (ou de flottants), chaîne de caractères, etc.

Le **parcours** d'une séquence (ou **parcours séquentiel**) consiste à énumérer les éléments de la séquence et à **appliquer un même traitement** à chacun d'eux.

Le traitement de la séquence se compose de quatre parties algorithmiques distinctes :

- Le **traitement de l'élément courant**.
- L'**initialisation** du traitement : instructions d'initialisation du processus de traitement.
- La **terminaison** du traitement : instructions à effectuer une fois la séquence parcourue.
- Le **traitement** à effectuer dans le cas où la **séquence est vide** (s'il y a lieu).

Exemple :

L'un des traitements les plus courants sur les tableaux est la **recherche d'un élément** précis dans le tableau. Il s'agit de parcourir le tableau en cherchant un élément donné et stopper le parcours de la liste dès que l'on trouve l'élément en question (la première occurrence de l'élément), la valeur de retour est alors l'indice de l'élément dans le tableau. Si l'élément ne figure pas dans le tableau l'algorithme rend par exemple -1.

2. Activités élèves

Dans notre projet, nous proposons trois activités de parcours séquentiel :

- Activité 1 : Ecrire un algorithme de calcul de moyenne d'entiers stockés dans un tableau.
- Activité 2 : Ecrire un algorithme de recherche d'une occurrence dans un tableau d'entiers.
- Activité 3 : Ecrire un algorithme de recherche d'un extremum dans un tableau d'entiers.

Les élèves écriront ces algorithmes en langage Python sur ordinateur, puis chercheront à calculer leur complexité (en montrant que cette complexité est linéaire) et à prouver leur terminaison (avec des boucles « while »).