# **Complexité** : correction TD1

1. À part la vitesse, qu’est-ce qui pourrait servir à mesurer l’efficacité dans un contexte concret ?

Entre 2 universités à un concours donné l’Université qui a plus de lauréats est plus efficace.

1. Sélectionnez une structure de données que vous avez déjà vue, puis étudiez ses avantages et ses inconvénients.

Tableau : structure de données.

* + - Plusieurs données de même type.
    - Taille fixée à la création.
    - Espace mémoire contigu.
    - Éléments références par des indices.
    - Ajouter dans une place libre connaissant l’indice.
    - Ajouter dans une place libre : consiste à parcourir le tableau pour chercher l’indice de la case vide et insérer l’élément.

## Algorithme ajout\_element(e : element, T : tableau n element)

## {

## Var i : ajouter : booléen

## I🡸 1; ajouter🡸false

## Tant que i<= n, ajouter = false faire

## Si T[i] = NIL alors

## T[i] 🡸 e

## Ajouter 🡸 True

## Fin si

## i🡸 i+1

## Fin tant que

## }

* + - Modifier un élément en position ‘i’.

## Algorithme modifier\_element(e : element, T : tableau n element, i : entier)

## {

## T[i] 🡸 e;

## }

* + - Supprimer un élément en position ‘i’ : consiste à décaler vers la gauche d’une case de tous les éléments à partir de la position ‘i+1’.

## Algorithme suppression\_element(T : tableau de taille n, position : entier)

## {

## Pour i 🡸 position a (n-1) faire

## T[i] 🡸 T[i+1]

## Fin pour

## T[i+1] 🡸 NIL

## }

* + - Rechercher un élément {Ɵ(n)}

## Algorithme recherche \_element (V : element, T : tab n element) : booléen

## {

## Var i : entier ; trouve : booléen

## Début

## I 🡸 0

## Trouve 🡸 faux

## Répéter

## I 🡸 i +1

## Si T[i] = V alors

## Trouve 🡸 vrai

## Jusqu’à  (i=n) ou (trouve)

## Return trouve

## }

Listes chainées simple : structure de données linéaires.

* + - Les données sont sous forme de séquence.
    - Une séquence contient un élément et l’adresse de la séquence suivante.

L

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 |  |  |  |  |  |  |  |  |  |

NULL

* + - Ajouter un élément :
* Enfin de lister parcourir les éléments jusqu’au dernier Ɵ(n).
  + - Modifier un élément en position ‘i’.