P31

Structures de données

# Vector :

* Zone mémoire contiguë
  + @début
  + Taille de l’élément
  + Nb d’éléments
* Réserver la mémoire à l’avance
  + Capacité > Taille effective
* Ajouter éléments
  + Vérifier que taille + 1 <= capacité
  + Ajout fin
    - v[taille] = e
    - taille++
  + Ajout i
    - Décaler éléments dont l’indice >= i
    - v[i] = e
    - taille++

# Liste chainée

* Nœuds indépendants
* Chacun contient l’adresse du suivant

# Pile / Stack

* Push
* Pop
* Top
* First In Last Out

# File / Queue

* Enqueue
* Deqeue
* Head
* First In First Out

# Arbre binaire / Tas

# Dictionnaire (Clé, valeur)

* Vector
  + Recherche en temps linéaire
* Arbre binaire de recherche
  + Utiliser une relation d’ordre sur les éléments (clés)
  + Recherche en log2(n) si l’arbre est équilibré
* Tableau
  + Clés d’entiers [0, 232-1]
    - Tableau très creux
    - Tableau très gros
      * Gaspillage
  + Clés ∈ k
    - buckets [0, p]
    - t : k -> [0, p]
    - h o r
    - hash réduction
    - k -> [0, 232-1] [0, 232-1] – [0, p]
    - p < 232-1
    - t(k) -> index
    - recherche -> 1
    - Si K > p
      * Collisions
    - Mécanismes de gestion
      * Ecraser
      * Chainage
    - Il faut que **H** soit **uniforme** (tous les éléments ont la même probabilité de sortir)