

# Table de hachage et dictionnaire

Quentin Fortier

October 15, 2021

On peut voir un tableau  $t$  comme une fonction qui à chaque indice  $i$  associe  $t.(i)$

Les indices sont forcément des entiers consécutifs à partir de 0.

On peut voir un tableau  $t$  comme une fonction qui à chaque indice  $i$  associe  $t.(i)$

Les indices sont forcément des entiers consécutifs à partir de 0.

Un **dictionnaire** est une structure de donnée plus générale qui à chaque **clé** associe une ou plusieurs **valeur(s)**. Il possède les opérations suivantes :

- Ajouter une association (clé, valeur)
- Supprimer une association (clé, valeur)
- Obtenir les valeurs associée à une clé donnée

Type abstrait impératif de dictionnaire en OCaml :

```
type ('k, 'v) dict = {  
    add : 'k * 'v -> unit;  
    del : 'k * 'v -> unit;  
    get : 'k -> 'v list  
}
```

get k renvoie [] si k n'est pas dans le dictionnaire.

## Dictionnaire : implémentation par liste de couples

On pourrait implémenter un dictionnaire avec une liste de couples (clé, valeur).

## Dictionnaire : implémentation par liste de couples

On pourrait implémenter un dictionnaire avec une liste de couples (clé, valeur).

```
let dict_of_list () =  
  let l = ref [] in {  
    add = (fun c -> l := c::!l);  
    del = (fun c -> l := List.filter ((=) c) !l);  
    get = (fun k ->  
      List.filter (fun (k', _) -> k'=k) !l  
      |> List.map snd)  
  };;
```

## Dictionnaire : implémentation par liste de couples

On pourrait implémenter un dictionnaire avec une liste de couples (clé, valeur).

```
let dict_of_list () =  
  let l = ref [] in {  
    add = (fun c -> l := c::!l);  
    del = (fun c -> l := List.filter ((=) c) !l);  
    get = (fun k ->  
      List.filter (fun (k', _) -> k'=k) !l  
      |> List.map snd)  
  };;
```

|> est le symbole de *pipe* :  $e \mid\!> f$  est équivalent à  $f\ e$ .

# Table de hachage

Une table de hachage est constituée:

- 1 d'un tableau (dynamique)  $t$  contenant les valeurs
- 2 d'une fonction de hachage  $h$  de l'ensemble des clés vers les indices de  $t$

La valeur associée à une clé  $c$  est stockée à l'indice  $h(c)$  du tableau  $t$ .



# Table de hachage

Une table de hachage est constituée:

- 1 d'un tableau (dynamique)  $t$  contenant les valeurs
- 2 d'une fonction de hachage  $h$  de l'ensemble des clés vers les indices de  $t$

La valeur associée à une clé  $c$  est stockée à l'indice  $h(c)$  du tableau  $t$ .

Si la même clé est associée à plusieurs valeurs alors  $t$  doit être un tableau de listes.

# Table de hachage

Une table de hachage est constituée:

- 1 d'un tableau (dynamique)  $t$  contenant les valeurs
- 2 d'une fonction de hachage  $h$  de l'ensemble des clés vers les indices de  $t$

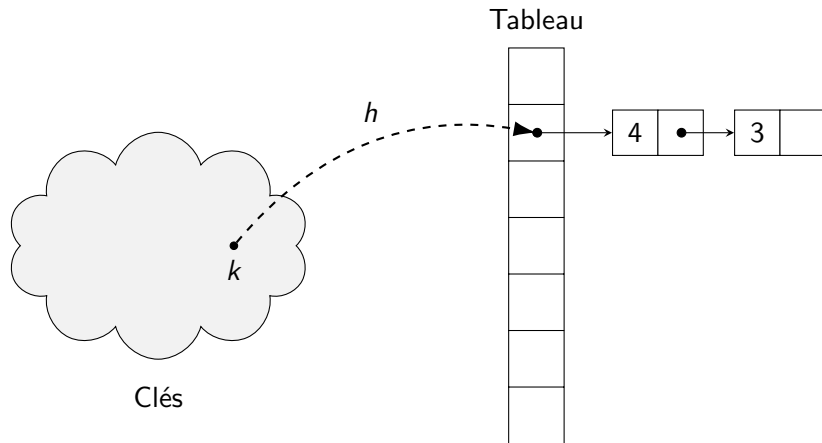
La valeur associée à une clé  $c$  est stockée à l'indice  $h(c)$  du tableau  $t$ .

Si la même clé est associée à plusieurs valeurs alors  $t$  doit être un tableau de listes.

Si les clés sont des entiers (non consécutifs), on peut choisir  
 $h : x \mapsto x \bmod n$ .

Sous quelques hypothèses, on peut montrer que les opérations de table de hachage sont en complexité moyenne  $O(1)$ .

# Table de hachage



$h$  associe à chaque clé  $k$  un indice du tableau, dans lequel est stockée la liste des valeurs de  $k$