

# Tema 3. Problema 1: El Hash tancat

Esteve Brugulat

Josep M. Ribó

5 de desembre de 2009

## 1 Objectius

- Comprendre el funcionament dels algorismes d'inserció i consulta en un hash tancat.

Per fer aquest problema, cal que us llegiu:

- Els apartats dels apunts: 3.4.1, 3.4.2 i 3.4.3

## 2 Enunciat

Es tracta de completar les operacions:

- **(Obligatori)** `put(k,v,it)`

```
template<class T,class FH, unsigned int B>
void HashMap<T,FH,B>::put(const MyString& k,
                        const T& x,
                        MapIterator<T>& iter)
    throw (FullMapException)
```

Respon en un document *respostes*, que inseriràs a la carpeta `docs` del projecte, les preguntes següents:

1. Cada node del hash té una marca que pot valdre: *sborrat*, *lliure* o *ocupat*. Per què és necessari el valor *esborrat*?
2. Quan, en un procés d'inserció d'una clau `k`, trobem una posició marcada com a *lliure*. Per què no podem inserir la clau `k` directament en aquella posició?
3. Què fa l'algorisme d'inserció si troba que la clau `k` que es vol inserir **ja és** a la taula?
4. Per a què serveix l'iterador que retorna l'operació `put`?

- **(Optatiu)** `get(k,v,found)`

```
template<class T, class FH, unsigned int B>
void HashMap<T,FH,B>::get(const MyString& k,
                        T& x,
                        bool& found) const
```

- **(Optatiu)** `remove(k)`

```
template<class T,class FH, unsigned int B>
void HashMap<T,FH,B>::remove(const MyString& k)
```