Gabriele Carrettoni - 746431 - g.carrettoni@campus.unimib.it

Set

Richieste

> Il progetto richiede la progettazione e realizzazione di una classe che implementa un Set di elementi generici T. Un Set è una collezione di dati che non può contenere duplicati: es. S= {1,6,4,9,7,10,12}. Implementare il Set in modo tale che l'accesso tramite un indice all'i-esimo elemento avvenga in tempo costante.

Implementazione

Inserimento

La classe controlla che l'elemento non esista già nel set ed in caso affermativo procede all'inserimento dello stesso nel set e nel filter, altrimenti throw l'eccezione alread_in. Nel caso in cui l'elemento da inserire sfori la grandezza del set, viene chiamata la funzione grow() che provvede ad aumentare la grandezza del set in modo esponenziale (*2) e a fare un realloc del puntatore.

Controllo

La classe controlla che l'elemento non sia già presente all'interno del set attraverso una query al filter passato all'inizializzazione della classe. Ci sono 4 esempi di filter nel file filter.h

BaseFilter: un semplice filter the risponde alla query per la presenza dell'elemento sempre con MAYBE, forzando così la classe Set a scorrere tutto il set alla ricerca dell'elemento.

BloomFilter: un counting bloom filter che risponde alla query per la presenza dell'elemento con MAYBE o NOT_FOUND. Essendo una struttura probabilistica è possibile ritorni falsi positivi, da qui la necessità di usare MAYBE nei casi la query sia positiva.

CuckooTable: una hashtable con cuckoo-hashing che contiene ogni elemento inserito nel set, risponde alla query per la presenza dell'elemento con NOT_FOUND o FOUND in O(1).

CuckooFilter: simile al bloom filter ma usando cuckoo partial hashing. Probabilistico come il bloom filter.

Eliminazione

La classe controlla che l'elemento sia presente, in caso affermativo lo elimina sia dal set sia dal filter passato. L'eliminazione dell'elemento, per evitare un

costoso reindex degli elementi successivi a quello eliminato viene utilizzato std::rotate per spostare l'elemento nella posizione last per poi invocare il rispettivo dtor e decrementare il range dei valori validi (--last).

Iteratori

Invece di implementare sia const_iterator sia iterator, Set.h provvede un _iterator con una variabile bool passata via template che determina se _iterator sia const o meno, questo evita la duplicazione del codice comune a const_iterator e iterator.