Name : Ahmed Mohamed Fathallah . ID : 14.

Lab#3 Report

Hashing Method class :

This class is responsible for the matrix universal hashing method .

Methods :

- The first method is create H which is given the number of rows needed in H matrix generates this number of random integers – 32 bits -.
- The second method is responsible for the hashing itself , given the array h of integers generated and the integer to be hashed , it performs matrix multiplication by bit masking . taking the binary AND of the current integer and the key then count the number of ones resulting , if its odd it produces one in the corresponding bit of the result hash index , else it produces zero . then it returns the result integer from this operation as the hash index .

Node Class :

Attributes :

It is a simple class representing the node holding the key , it has the integer key value and a boolean indicating whether it's a dummy node or not .

Methods :

- It has 2 methods , set which gives the key of the node a certain value and sets the dummy boolean by false .
- The second method is reset which deletes the value of the node and sets the dummy boolean by true.

Quadratic Table class :

It is the class producing the hash table of quadratic size .

Attributes :

- array h representing the hashing matrix .
- a hashing method object that will be responsible for generating H matrix and hashing keys .
- integer of number of rebuildings .
- boolean indicating if the table was built or not .
- array list representing the hash table .

Methods :

- build method which is given an array list of keys builds the hash table . it starts by filling the table list with dummy nodes of number equals the keys list size squared . then it takes the size needed for h matrix and generates one , the size needed is $\log_2$(table size) . after the h matrix is generated , it iterates on the keys list and use the hashing method object to hash each key , if the index returned causes a collision the table nodes are reset to dummy and a new h matrix is generated , number of rebuildings is incremented too . if no collision occurred then the hashing table is built (set to true) and the method ends .

- look up method responsible for searching for the existence of a key in the table . if the table was not built it returns false ,else the key is hashed using the h matrix generated during building the table . if the value of the resulting index in the table was equal to the key it returns true , else returns false .

- print stats method which is for verification of size of the table and the number of rebuildings .

Linear Table Class :

It is the class representing the hash table of linear size .

Attributes :

- h1 , the level 1 hashing matrix .
- hashing method object responsible for generating hashing matrices and doing the hashing operation .
- integer representing number of rebuildings .
- boolean representing whether the table is built or not .
- array of Quadratic table objects represeneting the hashed bins of the second and final level .

Methods :

- build method . at first it generates the first level hashing matrix with number of rows = $\log_2$(size of keys list). Then an array of array lists is filled with the hashing results of the keys list , even if collisions exist . the second level of the building process is excuted by building the array of Quadratic tables giving them the keys lists in each index of the array .number of total rebuildings is incremented by the number of rebuildings in each quadratic table built .

- look up method . if the table is not built it returns false , else the key is hashed by the h1 hashing matrix producing the level 1 index(ind1) . then it looks up in the quadratic table of index ind1 in the quadratic table array . if found returns true , false other wise.

- Prints stats method for verification . prints the total memory occupied by the hashing table , also prints the total number of rebuildings .

- Print single stats method for verification . prints the number of rebuildings for each quadratic table created for each bin .