Birla Institute of Technology & Science, Pilani
2nd Semester 2016-17 - CS F211 – Data Structures and Algorithms

Lab 8 – 18th March
Topics – Bit Vectors and Bloom Filters

Problem 1

Design and implement bit operations on a single *unsigned int* word as a bit vector:

- Let N be *sizeof(unsigned int)*. Then we can have a bit vector of size N represented by an integer word.
- Set operation: This operation sets the j^{th} bit (1<=j<=N) (counting 0 as the LSB) of the given integer word S to 1. (j is taken as input)
- Get operation. This operation returns the j^{th} bit (1<=j<=N) (counting 0 as the LSB) of the given integer word S. (j is taken as input)

Obtaining j^{th} bit of an *int* in C language can be achieved by masking the vector with a bit pattern and testing it: i.e. (S & B_j) iff (j^{th} bit of S is 1) where B_j is all 0s except the j^{th} bit.
B_j can be obtained by left-shifting *((unsigned int) 1)* j times.

Implement the set and get operations on an unsigned int, to be used as a bit vector. You can use the following table for designing your functions.

Key	Function	Input Format	Description
0	CreateBitVector	0	Creates a bit vector S in the form of an integer word of size – sizeof(unsigned int), which is equal to 32; And then initializes it to 0.
1	Set	1 j	Sets the value of j^{th} bit of S to 1.
2	Get	2 j	Returns the value of j^{th} bit of S. You must also print the returned value.

Sample input and output

Sample Input	Sample Output
0 1 2 1 6 1 9 2 6 2 9 2 31 -1	1 1 0