Birla Institute of Technology & Science, Pilani

2nd Semester 2016-17 - CS F211 – Data Structures and Algorithms

Lab 8 – 14th March

Topics – Sets and Hashtables

Problem 3

When the set $S \subseteq U$ is sparse, i.e. $|S| << |U|$, then space is wasted. (Why?)
One alternative is to use a smaller array and hash the element (i.e. j) into a location instead of using j as the index. Hashing also removes the restriction that elements of U be integers (so that can be used as indices). So, for any known universe U, we can hash its elements into a table of bits (to *add* or *find*), given a good hash function on U. Implement ADT HashSet which is same as Set but implemented as a hashtable instead of the array.

Assume you have set of integers. Let M be the size of the hash table which stores bit vectors (in the form of unsigned int) as its elements. You can use the following hashing function to find the index in the hash table:

For any $j \in U$, $H(j) =: |A * j + B|$ mod M, where A=19 and B = 23

And then the below property will be true.

Given $S \subseteq U$, for any $j \in U$, $(j \in S)$ iff ((j%N)th bit of S[H(j)] is 1).

In this scheme, you can't afford to have collisions while creating the set. You can assume that the hash function given doesn't give any collisions. Implement typical set operations *union*, *intersect*, and *difference* for this HashSet ADT. You can use the following table for designing your functions.

Key	Function	Input Format	Description
0	readData	0 M N X Y A_1 A_2 A_3 .. A_x B_1 B_2 B_3 .. B_y	M represents the size of the hash table. N represents the size of unsigned int (taken as input for convenience). X and Y represent the sizes of two sets of A & B respectively. You shall need to read two sets (A & B) of integers, separated by a new line. Each set contains values with space separation. Represent A and B in the form of SmallSet ADT described above.
1	Union	1	Perform union operation on A and B ($C = A \cup B$) and print C. You may sort C in ascending order first before printing.
2	Intersection	2	Perform intersection operation on A and B ($C = A \cap B$) and print C. You may sort C in ascending order first before printing.
3	Difference	3	Perform difference operation on A and B (C = A - B) and print C. You may sort C in ascending order first before printing.

Sample input and output

Sample Input	Sample Output

0 10 32 5 6
2 4 92 34 74
9 37 45 10 34 58
1
2
3
-1
	2 4 9 10 34 37 45 58 74 92
	34
	2 4 74 92