Hashing sets

Your company has a database $S \in U$ of keys. For this database, it uses a randomly chosen hash function h from a universal family H (as seen in class); it also keeps a bit vector B_S of m entries, initialized to zeroes, which are then set $B_S[h(k)] = 1$ for every $k \in S$ (note that collisions may happen). Unfortunately, the database S has been lost, thus only B_S and h are known, and the rest is no more accessible. Now, given $k \in U$, how can you establish if k was in S or not? What is the probability of error? Under the hypothesis that $m \geq c|S|$ for some $c > 1$ (note: we do not know the actual values of c and $|S|$) can you estimate the size $|S|$, i.e. the size of S, looking at just h and B_S? What is the probability of error? Note that S is no more accessible as it disappeared.
Optional: Another database R has been found to be lost: it was using the same hash function h, and the bit vector B_R defined analogously as above. Using h, B_S, and B_R, how can you establish if k was in $S \cup R$ (union), $S \cap R$ (intersection), or $S \setminus R$ (difference)? What is the probability of error?
SOLUTION