

Hash tables

Exercises

1. Suppose we have a dynamic set, S , stored in a direct-address table, T , of length m . Write a procedure for finding the maximum element in S . What is the worst-case running time of this procedure?
2. A **bit vector** is simply an array of bits. A bit vector of length m takes much less space than an array of m pointers. How could you use a bit vector to represent a dynamic set of distinct elements with no satellite data? Write dictionary operations for this implementation and confirm that their running times are all $O(1)$.
3. Suppose a hash function, h , hashes n distinct keys to an array T of length m . Assuming simple uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of $\{\{k,l\}: k \neq l \text{ and } h(k) = h(l)\}$
4. Show what happens when we insert the keys 5, 28, 19, 15, 20, 33, 12, 17 and 10 into a hash table where collisions are resolved by chaining. Suppose the table has 9 slots and let the hash function, $h(k) = k \bmod 9$.