Reading

See webpage.

Exercises

1 Run by Hand and Properties

1.1 $[w]$ Insert the key sequence $K = 7, 18, 2, 3, 14, 25, 1, 11, 12, 1332$ into a hash table of size 11 using chained hashing with hash function $h(k) = k \mod 11$.

1.2 $[w]$ Insert the key sequence $K = 7, 18, 2, 3, 25, 1, 11, 12$ into a hash table of size 11 using linear probing with hash function $h(k) = k \mod 11$.

1.3 $[w]$ Let K be a sequence of keys stored in a hash table A using chained hashing. Given A, can one efficiently find the maximum element in K?

1.4 $[w]$ Show the resulting hash tables from 1.1 and 1.2 after deleting key 2.

2 Streaming Statistics An IT-security friend of yours wants a high-speed algorithm to count the number of distinct incoming IP addresses in his router to help detect denial of service attacks. Can you help him?

3 Multi-Set Hashing A multi-set is a set M, where each element may occur multiple times. Design an efficient data structure supporting the following operations:

- ADD(x): Add an(other) occurrence of x to M.
- REMOVE(x): Remove an occurrence of x from M. If x does not occur in M do nothing.
- REPORT(x): Return the number of occurrences of x.

4 Lazy Deletion in Linear Probing Consider the following "lazy" strategy for deletion in linear probing. When an element is deleted at position p, we place a *tombstone* marker (e.g., a special value not in U) at p to indicate that the element is deleted.

4.1 Explain how SEARCH and INSERT should be modified to work when using this strategy.

4.2 Explain benefits and drawbacks using this method compared to "eager" deletion.

5 Quicksort Consider the following sorting algorithm: Construct a random permutation of the numbers and insert them in this order into an initially empty binary search tree. When all numbers are inserted, output the inorder sequence of the numbers. The search tree does not perform any rotations. To insert a number, search down to find the correct place and insert it as a leaf. I.e., inserting the numbers $1, 2, 3, 4, 5$ in the order $2, 1, 3, 4, 5$ would give the following search tree:

```
   2
  / \
 1   3
      \
       4
        \
         5
```

Prove that the expected running time is $O(n \log n)$. *Hint:* Argue that this is just another way of describing the Quicksort algorithm.

6 Dynamic Arrays and Dictionaries Consider a dictionary implemented using chained hashing combined with universal hashing. The array in the data structure has length m and the size of the stored set of elements is n. In the lecture, we require that $m = \Theta(n)$, but what happens if we insert a lot of elements into the dictionary so that this is no longer true? Solve the following exercises.

6.1 Suppose that $n \gg m$. Analyze the space and running time of operations.

6.2 Show how to modify the dictionary to handle sets that grow significantly while maintaining compact space and fast operations. Specifically, consider the scenario in which we start with an empty dictionary and repeatedly insert elements into it. Analyze the space and running time of operations. *Hint:* Think doubling arrays.

7 The Rabbit Billy (Exam 2017) The rabbit Billy lives with his family in a rabbit hole. The rabbit Billy is very forgetful, and he loves carrots. Last week, he hid k carrots in k different bushes. Now he is hungry, but he forgot in which of the b bushes around the rabbit hole he hid the carrots. To find a carrot, he now does the following. In each round, he goes to a random bush and checks if there is a carrot. If not, he runs back to the rabbit hole and tries again. Since he is very forgetful, he might go to the same bush again in the next round.

7.1 What is the expected number of bushes that Billy visits before he finds the first carrot? Explain your answer.

7.2 After eating the first carrot, Billy is still hungry, so he keeps looking for two more carrots. He does not remember where he has already found carrots, so he might go to these bushes again. What is the expected number of bushes that Billy visits before he has found three carrots? Explain your answer.