Exercise 1

Exercise: (CLRS 11.1-1) Suppose that a dynamic set S is represented by a direct-address table T of length m. Describe a procedure that finds the maximum element of S. What is the worst-case performance of your procedure.

Solution:

```
hashTableMax(t)
    max = t[1]
    for i = 2 to t.length
        if max < t[i]
            max = t[i]
    return max
```

Since the hash table is direct addressing it works much like an array, meaning the is no form of collision, thus we can simply go through the values, remembering that empty spots will contain null. Since the hash table is not sorted we have to go through the whole table once, meaning a worst case performance of $\Theta(n)$, where n is the length of the hash table.

Exercise 2

Exercise: (CLRS 11.2-2) Demonstrate what happens when we insert the keys 5; 28; 19; 15; 20; 33; 12; 17; 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be h(k) = k mod 9.

Solution: When we insert we insert at the head of the table, thus we get the following.

slot	Chain
0	null
1	10 → 19 → 28
2	20
3	12
4	null
5	5
6	33 → 15
7	null
8	17

Exercise 3

Exercise: (CLRS 11.2-5) Suppose that we are storing a set of n keys into a hash table of size m. Show that if the keys are drawn from a universe U with |U|

> nm, then U has a subset of size n consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is $\Theta(n)$. Hint: The Dirichlet's box principle —a.k.a. pigeon hole principle— states that for $n, m \in N$, if nm+1 objects are distributed among m sets, then at least one of the sets will contain at least n + 1 objects.

Solution: Given the relation $|U| > nm$ we now that $|U| >= nm + 1$, assuming even distribution in the m slots there, we can distribute n times and have 1 left over. Meaning that at least one chain will have $n + 1$ elements, with unique keys, this is the subset that of keys that all hash to the same value. Since in there is always a chain with at least $n + 1$ elements the worst case running time will be $\Theta(n)$.

Exercise 4

Exercise: (CLRS 11.4-1) Consider inserting the keys 10; 22; 31; 4; 15; 28; 17; 88; 59 into a hash table of length m = 11 using open addressing with the auxiliary function $h'(k) = k$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \text{ mod } (m - 1))$.

Solution: - Linear probing:

`$h(k, i) = (h'(k) + i) \text{ mod } m = (k+i) \text{ mod } 11$`

- **Quadratic probing:**

  $h(k, i) = (h'(k) + c_1 i + c_2 i^2) \text{ mod } m = (k + i + 3i^2) \text{ mod } 11$

- **Double hashing:**

  $h(k, i) = (h_1(k) + i h_2(k)) \text{ mod } m) = (k + i(1 + (k \text{ mod } 10)) \text{ mod } 11$

Apparently did not implement them correctly in Haskell so i am not finishing the table, but just follow the formulas starting from $i = 0$ incrementing i by 1 if there is a collision.

Index	Linear	Quadratic	Double
0	22	22	null
1	88	null	null
2	null	null	null
3	null	17	null
4	4	4	null
5	15	null	null
6	28	28	null
7	17	null	null
8	59	15	null
9	31	31	null

Index	Linear	Quadratic	Double
10	10	10	null

Exercise 5

Exercise: (CLRS 11.4-3) Consider an open-address hash table with uniform hashing. Give upper bounds on the expected number of probes in an unsuccessful search and on the expected number of probes in a succesful search when the load factor is 3/4 and when it is 7/8.

Solution: By theorem 11.6 unsuccessful search is $\frac{1}{1-\alpha}$, thus we get $\frac{1}{1-\frac{3}{4}} = 4$ and $\frac{1}{1-\frac{7}{8}} = 8$

By theorem 11.8 succesful search is $\frac{1}{\alpha} ln(\frac{1}{1-\alpha})$, thus we get $\frac{1}{\frac{3}{4}} ln(\frac{1}{1-\frac{3}{4}}) \approx 1.85$ and $\frac{1}{\frac{7}{8}} ln(\frac{1}{1-\frac{7}{8}}) \approx 2.38$