Week 3. Problem set (solutions by Danil Elgin)

1. Consider a hash table with 13 slots and the hash function $h(k) = (k^2 - k + 11) \mod 13$. Show the state of the hash table after inserting the keys (in this order)

$$6,\ 28,\ 19,\ 15,\ 20,\ 33,\ 12,\ 17,\ 10,\ 13,\ 3,\ 39$$

with collisions resolved by *linear probing* [CLRS, §11.4].

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Key	28	15	6	19	20	33	12	3	39		17	10	13

Answer: *In the table*
Justification (optional): *We substitute each number into the hash function h(k) = (k^2−k+11) mod 13 and, if a cell in the hash table is occupied, add +1 to the index until we find an empty cell or the addition amount is 13 - 1.*

2. Consider a dictionary that maps an integer (e.g. group number) and a short[1] string (e.g. student email) to a number (e.g. grade), and is implemented as a hashtable T of hashtables:

- for an integer k, if T contains k, then $T[k]$ is a hashmap with string keys
- for a string s, if $T[k]$ contains s, then $T[k][s]$ is a floating point number

Compute average case time complexities of successful and unsuccessful search in this hashtable of hashtables with the different possible combinations of chaining and open addressing, assuming

(a) *independent uniform hashing* for hashtables with separate chaining [CLRS, §11.2]
(b) *independent uniform permutation hashing* for hashtables with open addressing [CLRS, §11.4]
(c) for the hashtable T, load factor α and size n
(d) for the hashtables $T[k]$, average load factor β and average size m

	Successful search	Unsuccessful search
T uses separate chaining and each $T[k]$ uses separate chaining	$O(1+\alpha+\beta)$	$O(1+\alpha+\beta)$
T uses separate chaining and each $T[k]$ uses open addressing	$O(1+\alpha+\frac{1}{\beta}\ln(\frac{1}{1-\beta}))$	$O(1+\alpha+\frac{1}{1-\beta})$
T uses open addressing and each $T[k]$ uses separate chaining	$O(1+\beta+\frac{1}{\alpha}\ln(\frac{1}{1-\alpha}))$	$O(1+\beta+\frac{1}{1-\alpha})$
T uses open addressing and each $T[k]$ uses open addressing	$O(\frac{1}{\alpha}\ln(\frac{1}{1-\alpha})+\frac{1}{\beta}\ln(\frac{1}{1-\beta}))$	$O(\frac{1}{1-\beta}+\frac{1}{1-\alpha})$

3. **(+0.5% extra credit)** Compute the average case time complexity of the following algorithm:

```
secret(M, k):
  Q := new linked queue
  while M.hasKey(k) and k > 0
    item := M.get(k)
    k := min(item.value, k) - 1
    if k == item.value - 1
      Q.offer(item)
  while not Q.isEmpty()
    M.remove(Q.poll())
  return k
```

M is a hashtable of size n with load factor α that resolves collisions by *separate chaining* [CLRS, §11.2]. The answer **must** use O-notation and may depend on n, k, and α. Assume *independent uniform hashing* and worst case for the contents (values) of the input map M.

Justify your answer (3–5 sentences). Detailed proof is not required.

[1] Assume that strings have a relatively small constant maximum size.

Answer:$O((1+\alpha)*(k+1))$
Justification: The first while loop cost $(1+\alpha)*(k+1)+k+1$**. If we assume that we have the worst case scenario for the contents of the input map M, then at each iteration we will have** $k = item.value - 1$**. This means that we would repeat the while loop status check k+ 1 times. To verify that** $M.hasKey(k)$**, we need to take the average time complexity of** $1+\alpha$ **to search through a map that resolves collisions using a separate chain. To check the condition k ¿ 0, we need constant time. Finally, we get** $O(1+\alpha)*(k+1)+k+1$, but we know that $1+\alpha > 1$, which means we can just take $(1+\alpha)*(k+1)$.. All other operations are either equal or cost less than the cost of the first while loop.

References

[CLRS] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2022. *Introduction to algorithms, Fourth Edition.* MIT press.