PS4

June 24, 2025

Problem 4-1

(a)

Summing the words won't match the expeted value shown in the lecture , since reordering the words in a string will not produce a different hash.

(b)

Both are required to maintain performance and correctness. There is always a chance of collision, so in order to maintain correctness one requires collision resolution. The performance would be harmed without dynanic resizing, which reduces the probablity of collision.

(c)

If Alyssa is enlarging a table of size m into a table of size m' and the table contains n elements, we need to rebuild the hash table, which is $O(m'+n)$. Initiating the hash table of size m' takes $O(m')$ and going over the slots of the old table and copying each item takes $O(m+n)$. Since $m' > m$, we have $O(m'+n)$

(d)

In dynamical resizing if instead of taking $m' = 2m$ we take $m' = m + k$, the method becomes inefficient. This method is similar in scaling to the method of enlarging the hash table each time by one, which scaled as $O(1+2+3+\cdots n) = O\left(n^2\right)$. Increasing by a constant number of each time doesn't modify the scaling of the algorithm:

$$O(k(1+2+\cdots+n)) = O\left(kn^2\right) = O\left(n^2\right)$$

In practice, doubling the size of the table is benificial due to the binary representation of all the computer's memory is based on.

Problem 4-2

(a)

Within the Principal Use Cases for Dictionaries page, the case of membership testing we expect: many insertions right after creation and then mostly lookups. As you create some new group and introduce a membership, you add all the members initially and then expect that the members don't change much, but the membership checks are frequent.

(b)

Membership testing benifits relatively sparce hash tables in order to reduce collisions (efficient lookups) and allow for fast open addressing (empty buckets allow fast open addressing). As a result, taking a large minimum size and a growth rate of 4 will lead to a relatively sparse hash table.

Problem 4-3

The code is implemented in dnaseq.py., it can be tested using test_dnaseq.py. The supporting files have been modified in order to run on Python3.