Problem Set 4

Problem 4-1

(a) Answer: Allysa's protocol will result in more collisions than in simple uniform hashing, because several permutations of the same characters or words will result in the same hash value. For example, "Man bites dog" and "Dog bites man" will hash to the same value. Answer is 3.

(b) Answer: To ensure performance and correctness you need both collision resolution and dynamic table resizing. Dynamic resizing maintains performance by keeping the expected number of collisions at $O(1)$. But collision resolution is still necessary to ensure correctness when collisions do occur. Answer is 4.

(c) Answer: Allocating space for new table of size m' is $\theta(m')$. Rehashing existing keys into new table requires looping through m tables entries that contain n keys in total, which takes $\theta(m+n)$. Total time is therefore $\theta(m+m'+n)$. Because $m' > m$, this simplifies to $\theta(m'+n)$.

(d) Answer: This is an improvement over re-sizing the table with every update, but the number of updates for a table of size n is till $O(n/k) = O(n)$, and total costs are still $O(n^2)$, because $\sum_{i=0 mod k}^{n} i = \frac{n}{k} \cdot \frac{n}{2} = O(n^2)$.

Problem 4-2

(a) Answer: This use case calls for the initial creation of a table, followed (mostly) by lookups. Answer is 1.

(b) Answer: Dictionaries for membership testing can be of "any size", so a large minimum size could potentially waste a lot of room for small tables. A growth rate of 2 - table doubling - can realize performance gains in a base-2 architecture, so answer is 2 - a small minimum size and a growth rate of 2.

Problem 4-3.

See code at https://github.com/claytonm/6006/blob/master/ps4/dnaseq.py.