2) (10 pts) ANL (Hash Tables)

This question asks you to explore the **best-** and **worst-case** runtimes for adding *r* new elements to a hash table that already contains *q* elements. In answering the questions below, assume the following:

1. Generating the initial hash value for any given key takes O(1) time.
2. We are using quadratic probing.
3. Our hash table is at least half empty, and the length of the table is prime.
4. There is enough space in the hash table to allow for all *r* new elements to be inserted without triggering a table expansion.
5. If the specific placement of the *q* elements that are already in the hash table is relevant, you may assume that they are placed in a way that would facilitate the situation you are describing that leads to the best- or worst-case runtime for the *r* new elements being added to the table.

Note that this question is not just asking for the runtime for adding a single element. We want the runtime for adding **all** *r* elements to the hash table. While the *q* elements already in the table may impact the runtime for adding the *r* new elements, you do not have to account for the runtime it took to add those *q* elements. Focus only on the cost of adding the *r* **additional** elements.

a. (2 pts) In big-oh notation, what is the **best-case** runtime for adding *r* new elements to the table?

O(r)

b. (1 pt) What situation leads to the best-case runtime you listed in part (a)?

We encounter no collisions.

c. (5 pts) In big-oh notation, what is the **worst-case** runtime for adding *r* new elements to the table?

$O(qr + r^2)$ --or-- $O(r(q + r))$

d. (2 pts) What situation leads to the worst-case runtime you listed in part (c)?

All *r* elements map to the same initial index in the hash table, **and** the first of those elements collides with all *q* elements already in the hash table before finding an open position (meaning that all subsequent elements will collide with all *q* elements in the table and any of the *r* elements that have already been inserted). The overall number of collisions is:

$$rq + \sum_{i=0}^{r-1} i = rq + \frac{(r-1)r}{2} = rq + \frac{r^2 - r}{2} = O(rq + r^2)$$

See grading notes on following page.

Grading for Question #2:

- **a.** +2 points for correct answer. This part is all or nothing (no partial credit).
- **b.** +1 point for correct answer.
  +5 points for $O(qr + r^2)$, $O(r(q + r))$, $O(r*\max(r,q))$
  +3 points for $O(qr)$ or $O(r^2)$
  +2 points for $O(qr^2)$, $O(q^2)$
  +1 point for $O(r\lg r)$, or any variation with r or q with a linear and log factor, or for $O(r)$ or $O(q)$.
- **c.** This part is worth two points total, as follows:
  - +1 point for mentioning the collision with all *q* elements
  - +1 for saying all *r* elements map to the same initial position in the table --or-- that each of the *r* new elements collides with all of the other *r* elements that were inserting before it.