Computational Thinking
Exercise 7 (Hashing)

1 Robin Hood Probing

In hashing with probing we notice that some objects can be placed in buckets early in their probing sequence while others may travel longer in their probing sequence until they find an empty bucket. Robin Hood hashing is trying to rectify this unfair situation by taking from the "short" and giving to the "long". Specifically, we make the following modification in the insertion process of hashing with linear probing: If the bucket is already occupied, then the object that has traveled longer in its probing sequence remains in the bucket.

a) How does this change affect the expected value of the probe sequence length?

b) How does this change the expected value of the longest probe sequence length?

c) How does this change affect the variance of the probe sequence length?

2 Hashing with Probing: Deletion

Consider how to delete entries from a hashtable with probing.

a) What goes wrong if our delete operation simply searches for a key and deletes it if found?

b) Suggest how to simply implement the delete operation (you can modify search slightly).

c) Suppose a lot of keys are placed in the hash table and then almost all are removed. Does the search performance deteriorate with your implementation? Explain the problem.

d) Suppose the probing is linear. Suggest a delete operation that improves the search time after a lot of keys are removed. *Hint: Think about shifting some objects.*

3 Not Quite Universal Hashing

Recall the universal family from the script: $\mathcal{H} := \{h_a : a \in [m]^s\}$ where $h_a(k_0, \ldots, k_{s-1}) = \sum_{i=0}^{s-1} a_i \cdot k_i \mod m$ for some prime m. Show that if we restrict the a_i to be nonzero and $s > 1$, $m > 2$, then $\mathcal{H}$ is no longer a universal family. *Hint: Find two keys with a collision probability of more than* $\frac{1}{m}$.