Spencer Shimko
CMSC 441
HW 5

1. 11.2-1
Suppose we use a hash function *h* to hash *n* distinct keys into an array of length *m*. Assuming uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of {{*k* , *l*} : *k* != *l* and *h*(*k*) = *h*(*l*)}?

X = { 1 if collision, 0 otherwise }

E[X] = 1/m; Y = number of collisions

E[Y] = E[Σ X] = Σ E [X]
E[Y] = Σ E[X] = Σ (1/m) = picktwo (2) (1/m)
E[Y] = n(n-1)/2^m

2. 11.4-4
Consider an open-address hash table with uniform hashing. Give upper bounds on the expected number of probes in an unsuccessful search and on the expected number of probes in a successful search when the load factor is 3/4 and when it is 7/8.

I was unsure of this question. Did it want a proof of the upper bounds with the stated load factor or did it want us to simply calculate the upper bound?

α = *n/m*
Unsuccessful: E[Xu] = 1 / (1 – α)
Successful: E[Xs] = (1 / α) ln (1 / (1 – α))

α = 3/4 :
Unsuccessful = 1 / (1 - 3/4) = 4
Successful = (1 / 3/4) ln (1 / (1 – 3/4)) = 1.848392

α = 7/8 :
Unsuccessful = 1 / (1 – 7/8) = 8
Successful = (1 / 7/8) ln (1 / (1 – 7/8)) = 2.376504

3. 11.3-4
Consider a hash table of size *m* = 1000 and a corresponding hash function *h*(*k*) = floor(*m* (*k A* mod 1)) for *A* = (sqrt (5) – 1) / 2. Compute the locations to which the keys 61, 62, 63, 64, and 65 are mapped.

61: 700
62: 318
63: 936
64: 554
65: 172

Also try this for another value of *A*, and some other values of *n* and *m* if you like. Which value of *A* seems to "spread out" the hash values better? How would you measure this?

After sampling with a few values for *A*, *n,* and *m* it seems that (sqrt(5) - 1) / 2 spreads the values fairly well. You could measure this by calculating standard deviance across the hashed locations.