1: Expected Length of Coding Scheme

6 points
Suppose that several information sources generate symbols at random from a five-letter alphabet A,B,C,D,E with different probabilities. We will try different encoding schemes for encoding these symbols in binary. Given the probabilities and an encoding scheme, you will compute the expected length of the encoding of n letters generated by the information source. You may use any rigorous method of analysis you like, but must show your work and justify your answer,

(a) Suppose that the symbols occur with probabilities Pr[A] = 0.4, Pr[B] = 0.2, Pr[C] = 0.2, Pr[D] 0.1, and Pr[E] = 0.1 , and the coding scheme encodes these symbols into binary codes as follows:
A 001
B 010
C 011
D 100
E 101
What is the expected number of bits required to encode a message of n symbols?

By using the equation for $E[X_i]$, we can denote the following as:

$$\begin{aligned} E[X_i] &= 3 * Pr[X_i = 1] + 3 * Pr[X_i = 2] + 3 * Pr[X_i = 3] + 3 * Pr[X_i = 4] \\ &= 3 * 0.3 + 3 * 0.3 + 3 * 0.2 + 3 * 0.1 + 3 * 0.1 \\ &= 3n \end{aligned}$$

The length is thus 3n.

(b) Now suppose that the symbols occur with the same probabilities Pr[A] = 0.4, Pr[B] = 0.2, Pr[C] = 0.2, Pr[D] = 0.1, and Pr[E] = 0.1, but we have a different encoding scheme:
A 0
B 10
C 110
D 1110
E 1111
What is the expected number of bits required to encode a message of n symbols?

By using the equation for $E[X_i]$, we can denote the following as:

$$\begin{aligned} E[X_i] &= 1 * Pr[X_i = 1] + 2 * Pr[X_i = 2] + 3 * Pr[X_i = 3] + 4 * Pr[X_i = 4] + 4 * Pr[X_i = 4] \\ &= 1 * 0.3 + 2 * 0.3 + 3 * 0.2 + 4 * 0.1 + 4 * 0.1 \\ &= 2.3 \end{aligned}$$

The length is thus 2.3n.

c Now consider a different information system that generates symbols with probabilities Pr[A] =0.5 , Pr[B] = 0.3, Pr[C] = 0.1, Pr[D] = 0.05, and Pr[E] = 0.05. We will use the same

encoding scheme:
A 0
B 10
C 110
D 1110
E 1111
What is the expected number of bits required to encode a message of n symbols?

By using the equation for $E[X_i]$, we can denote the following as:

$$E[X_i] = 1 * Pr[X_i = 1] + 2 * Pr[X_i = 2] + 3 * Pr[X_i = 3] + 2 * Pr[X_i = 4] + 4 * Pr[X_i = 4]$$
$$= 1 * 0.5 + 2 * 0.2 + 3 * 0.2 + 4 * 0.05 + 4 * 0.05$$
$$= 1.9$$

The length is thus 1.9n.

2. Random Gene Sequences

6 points

Suppose a strand of DNA is generated by appending random nucleotides with equal probability from the set A,T,G,C to the end of the sequence. What is the expected length of the sequence needed to get two As in a row? You may use any rigorous method of analysis you like, but must show your work and justify your answer.

Given the set of nucleotides are 4, and each have an equal probability from the set A,T,G,C, we know that the probability of each of these elements in the set is:
$Pr[A] = 0.25, Pr[T] = 0.25, Pr[G] = 0.25, Pr[C] = 0.25$
To understand the expected length of the sequence needed to get two A's in a row, we must break down the probabilities of this subset as follows:
Probability of A is: $Pr[A] = 0.25$
Probability of All Other Elements in set: $Pr[T, G, C] = 0.75$
By the definition of Expected value, we have:
$E[X] = 1 * E[X_A = 0.25] + (1 + E) * E[X_TGC = 0.75] = 4$
Algebraically, this can be rewritten as:
$4E = 16 + 1 + 3 + 3E$
$= 20 + 3E$
By solving for E:
$E = 20$
$\therefore$ The expected length of the sequence needed to get two A's in a row is 20.

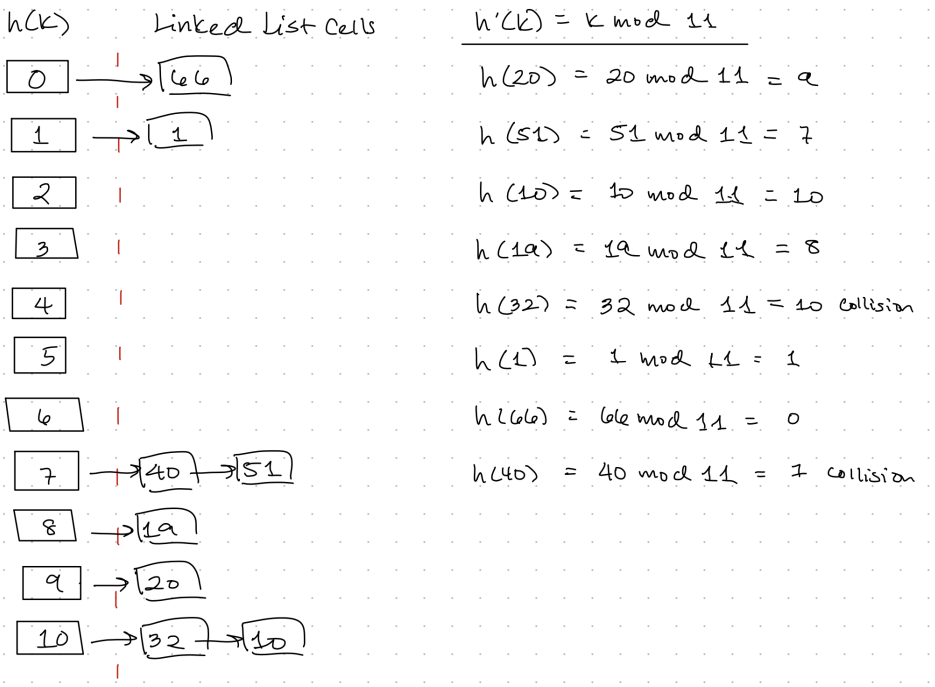
3. Hashing with Chaining

6 points

(a) (2 pts) Consider a hash table with m slots that uses chaining for collision resolution. The table is initially empty. What is the probability that, after k keys are inserted, there is a chain of size k? Include an argument for or proof of your solution.

The probability that after k keys are inserted there is a chain of size k is $\frac{1}{m^{k-1}}$. Because there is no reason to determine the position of the first set of key hashes, we can determine that for a chain to have a size k, the keys must hash into the same location - that is, there are an equal number of slots that pertains to the number of keys to fill those slots. Because the probability of a key that hashes into a location is $\frac{1}{m}$, and given that k-1 keys will hash into a given position, we know that the probability is not dependent on other values.

(b) (4 pts) Show the table that results when 20, 51, 10, 19, 32, 1, 66, 40 are cumulatively inserted in that order into an initially empty hash table of size 11 with chaining and h(k) = k mod 11. Use the Google Doc table below, with positions indexed 0 to 10, and linked lists going off to the right. Do not enter anything in cells that are not used.



4. Open Addressing Strategies

12 points

(a) (4 pts) Show the table that results when 20, 51, 10, 19, 32, 1, 66, 40 are cumulatively inserted in that order into an initially empty hash table of size 11 with linear probing

and
$h'(k) = kmod11$
$h(k,i) = (h'(k)mod11)$, where the first probe is probe i = o.
Draw this and the next result as horizontal arrays indexed from 0 to 10 as shown below. (You can fill in the Google Doc table.) Show your work in part (b) to justify your answer!

32	1	68	40				51	19	20	10
0	1	2	3	4	5	6	7	8	9	10

(b) (1 pt): How many re-hashes after collision are required for this set of keys? Show your work here so we can give partial credit or feedback if warranted.

There will be a total of 10 rehashes.
Given $h(k) = k$
h(20) = 20 $h(51) = 51$h(10) = 10 $h(19) = 19$h(32) 32 $h'(32,1) = (32+1)mod11 = 0$
$h(1) = 1mod11 = 1$
$h(66) = 66mod11 = 0$ REHASH
$h'(66,1) = (66+1)mod11 = 1$ REHASH
$h'(66,2) = (66+2)mod11 = 2$
$h(40) = 40mod11 = 7$ REHASH
$h(40,1) = (40+1)mod11 = 8$ REHASH
$h(40,2) = (40+2)mod11 = 9$ REHASH
$h(40,3) = (40+3)mod11 = 10$ REHASH
$h(40,4) = (40+4)mod11 = 0$ REHASH
$h(40,5) = (40+5)mod11 = 1$ REHASH
$h(40,6) = (40+6)mod11 = 2$ REHASH
$h(40,7) = (40+7)mod11 = 3$

(c) Show the table that results when 20, 51, 10, 19, 32, 1, 66, 40 are cumulatively inserted in that order into an initially empty hash table of size m = 11 with double hashing and
$h(k,i) = (h_1(k) + ih_2(k))mod11$
$h_1(k) = kmod11$
$h_2(k) = 1 + (kmod7)$
Refer to the code in the book for how i is incremented. Show your work in part (d) to justify your answer!

66		1	40	32			51	19	20	10
0	1	2	3	4	5	6	7	8	9	10

(d) (1 pt): How many re-hashes after collision are required for this set of keys? Show your work here so we can give partial credit or feedback if warranted.

There will be a total of 2 rehashes.

Given $h(k,i) = (h_1(k) + ih_2(k))mod11$, $h_1(k) = kmod11$, and $h_2(k) = 1 + (kmod7)$ appropriate hashing is shown below:

$h(51,0) = (51mod11 + 0(1 + (51mod7))mod11 = 7$
$h(10,0) = (10mod11 + 0(1 + (10mod7))mod11 = 10$
$h(19,0) = (19mod11 + 0(1 + (19mod7))mod11 = 8$
$h(32,0) = (32mod11 + 0(1 + (32mod7))mod11 = 10$ REHASH
$h(32,1) = (32mod11 + 1(1 + (32mod7))mod11 = 4$
$h(1,0) = (1mod11 + 0(1 + (1mod7))mod11 = 1$
$h(66,0) = (66mod11 + 0(1 + (66mod7))mod11 = 0$
$h(40,0) = (40mod11 + 0(1 + (40mod7))mod11 = 7$ REHASH
$h(40,1) = (40mod11 + 1(1 + (40mod7))mod11 = 2$

(e) (2 pts): Open addressing insertion is like an unsuccessful search, as you need to find an empty cell, i.e., to not find the key you are looking for! if the open addressing hash functions above were uniform hashing, what is the expected number of probes at the time that the last key (40) was inserted? Use the theorum for unsuccessful search in open addressing and show your work. Answer with a specific number, not O or Theta.

The specific expected number is 2.75.
As stated in the lecture, we have $\alpha = \frac{n}{m}$. Given this, we can substitute the values to be $\frac{1}{1-\alpha}$, as specified by the definition in class.
If the last key (40) was inserted, it is understood that 7 other inserted values are already inserted, and can be specified as $7.\dot{}$. The equation is such:

$$\frac{1}{1-\alpha} = \frac{1}{1-\frac{7}{11}}$$

It is proven then that 2.75 the specific expected number of probes.