Shruti Jadon, Abhiram Eshawaran, Hia Ghosh, Pardis Malekhzadeh

1. We need to find ~~$P\left(X \geq \frac{n}{2}\right)$~~ $P\left(X \geq \frac{n}{2}\right) \leq 0.001$

$$X_i = \begin{cases} 1 & \text{when we get a head} \\ 0 & \text{when we get a tail} \end{cases} \quad \Bigg| \quad \begin{array}{l} P(H) = \frac{1}{3} \\ P(T) = \frac{2}{3} \end{array}$$

$$E[X_i] = 1 \times \frac{1}{3} + 0 \times \frac{2}{3} = \frac{1}{3}$$

$$E[X] = E\left[\sum X_i\right] = \sum E[X_i] = \frac{n}{3}.$$

$$P\left(X \geq \frac{n}{2}\right) \leq 0.01$$

$$P\left(X \geq \left(\frac{n}{3} + \frac{n}{6}\right)\right) \leq 0.01 \Rightarrow P\left(\left(X - \frac{n}{3}\right) \geq \left(\frac{n}{3}\right) \times \frac{1}{2}\right) \leq 0.001$$

Applying Chernoff, $P\left((X - E[X]) \geq \frac{1}{2} \times E[X]\right) \leq 0.001$

$\therefore \delta = 1/2$

$$\therefore \quad P\left((X - E[X]) \geq \frac{1}{2} \times E[X]\right) \leq e^{-E[X]\delta^2/3}$$

$$e^{-E[X]\delta^2/3} \leq 0.001$$

$$\Rightarrow e^{-n/3 \times \frac{1}{4} \times \frac{1}{3}} \leq 0.001$$

$$\Rightarrow e^{-n/36} \leq 0.001$$

$$\Rightarrow -\frac{n}{36} \leq -6.9077$$

$$\Rightarrow n \geq 6.91 \times 36$$

$$\Rightarrow n \geq 248.6$$

$$\Rightarrow n \geq 249.$$ Ans

2)

Prob(a bit in the ith partition is $\emptyset$) = Prob(ith hash function maps none of m elements to this cell) $= \left(1 - \text{Prob}(\text{ith hash function maps element } j \text{ to this cell})\right)^m$

$$= \left(1 - \frac{1}{\frac{n}{k}}\right)^m = \left(1 - \frac{k}{n}\right)^m$$

Prob(a bit in the ith partition is 1) = 1 − Prob(a bit in the ith partition is $\emptyset$)

$$= 1 - \left(1 - \frac{k}{n}\right)^m$$

$\Rightarrow$ Prob(False Positive) $= \prod_{i=1}^{k}$ Prob(a bit in the ith partition is 1)

$$= \left(1 - \left(1 - \frac{k}{n}\right)^m\right)^k = \left(1 - e^{-\frac{m}{n/k}}\right)^k = \left(1 - e^{-\frac{km}{n}}\right)^k \quad \dots \text{①}$$

K hash functions into a single array:

$* =$ Prob(a bit is $\emptyset$) $= \prod_{i=1}^{k} \underbrace{\text{Prob}(\text{ith hash function maps no item to this cell})}_{\left(1 - \text{Prob}(\text{ith hash function maps item } j \text{ to this cell})\right)^m}$

$$\Rightarrow * = \prod_{i=1}^{k} \left(1 - \frac{1}{n}\right)^m = \left(1 - \frac{1}{n}\right)^{mk} \rightarrow \text{Prob}(\text{a bit is } 1) = 1 - \left(1 - \frac{1}{n}\right)^{mk}$$

$\Rightarrow$ Prob(False Positive) = Prob(All k cells corresponding to k maps are 1)

$$= \prod_{i=1}^{k} (\text{Prob ith hash function maps to a } 1) = \left(1 - \left(1 - \frac{1}{n}\right)^{mk}\right)^k = \left(1 - e^{-\frac{mk}{n}}\right)^k \quad \dots \text{②}$$

Therefore by ① & ②, the false positive rates are equal.

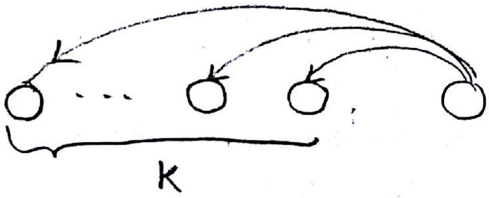
Scanned by CamScanner

Question 3: a) We construct a graph using items. For each item in the stream, make a node whose label specifies the item. For example, if there are n different items, and if ith item in the stream is of type j, create a node with label j for this item. Now we want to add some edges to this graph:

Consider the ith item of the stream:

- if it is the same as some item on the list, when incrementing its counter, also create a new node for it as described above
- if it differs from all items on the list:
  - if there are less than k items on the list, just create a node for it as described above.
  - else, along with decrementing current counters, when decrementing a counter, consider the time this counter was increased. For example, assume that a counter is 6 and you want to decrement it to 5. Consider the time when this counter was 5 and we increased it to 6. this increasing was because of observing a new item in the stream. Consider the node corresponding to this item. Make a new node as connect it with a directed edge to this node, such that this new node points to the previously discussed node.

At the end of the algorithm, the outer degree of each node is $\emptyset$ or k (because it has decremented k counters). Consider each node with outer degree k.



Note that inner-degree of each node is atmost 1. So we can partition graph nodes into groups of size (k+1), such that each partition consists of a node of outer degree k, alongwith k nodes it is connected to.

Now we prove the statement by contradiction. consider an element whose frequency is more than $\frac{m}{k+1}$, but it is not on the list. So each occurence of this item in the stream has decremented other counters or this occurence has been neutralized by some other item. $\Rightarrow$ each node whose label corresponds to this label is in a partition of size (k+1). $\Rightarrow$ the graph must have m ...

-3a- continued

than $(K+1) \times \frac{m}{K+1} = m$ nodes, which is a contradiction.

Therefore any item with frequency greater than $\frac{m}{K+1}$ will appear on list.

<u>Example</u> :- Let the stream be 1 2 3 4 5 6 1 2 3 1 2

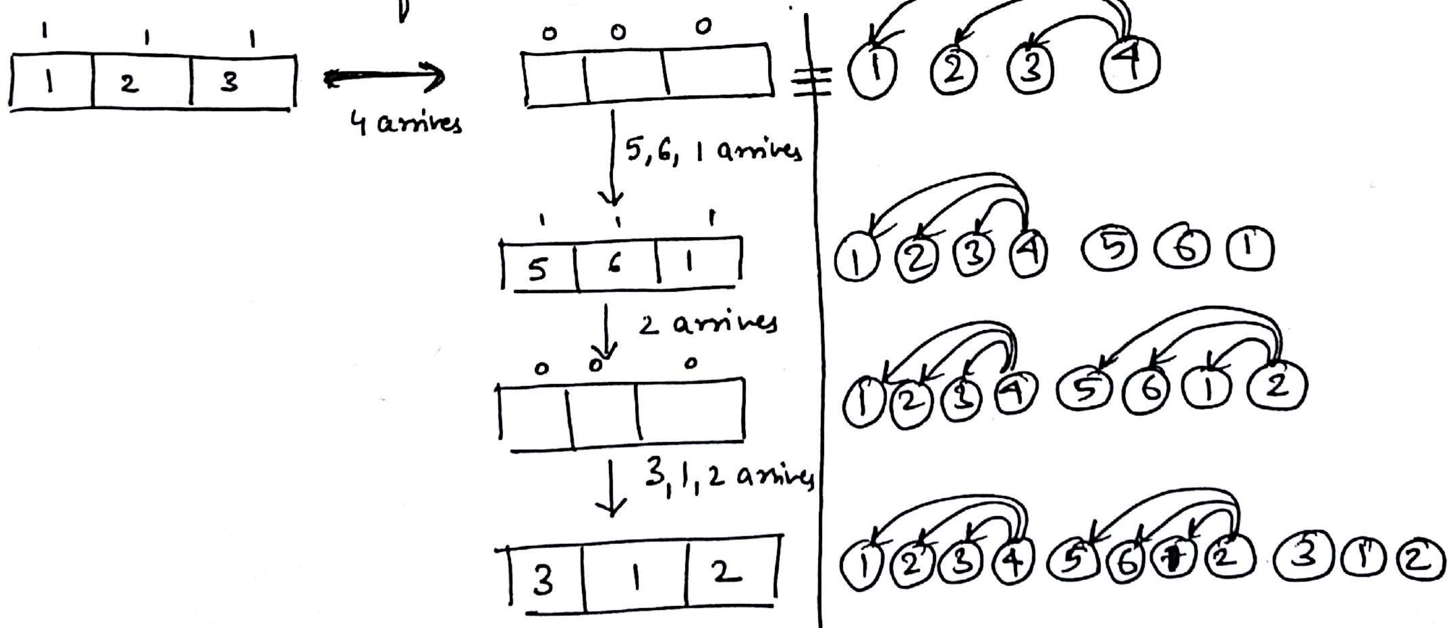
So, $m = 11$

Let us consider list of size, $K = 3$.

therefore, $\frac{m}{K+1} = \frac{11}{4} = 2.75$

So any element with frequency $= 2.75 \simeq 3$ should appear in the list.

~~Step~~1: In our illustration below, we form nodes for each incoming element. When the list becomes full, the count for the elements in the list is decremented. We draw edges from the incoming element to the last occurrence of the elements in the list.



'1' ~~and~~ '2' had a freq of $3 > 2.75$ and they are retained in the List.