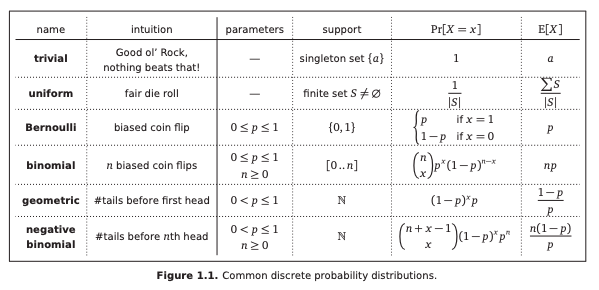
### Helpful Notes:



Suppose you play a game where you flip a fair coin until the sequence of flips satisfies

some condition. For each condition, compute the exact expected number of flips until that

condition is met

### Solution 1a.

you flip a heads.

**Method 1**: Since this is a fair coin and flipping a heads is a Geometric random variable, the expected value is 1/p = 1/(1/2) = 2

**Method 2**:

E(X) = P(H) \* 1 + P(T) \* (1 + E(X))

= 1/2 + 1/2 (1 + E(X))

E(X) = **2**

### Solution 1b.

you flip a heads and a tails

E(X) = P(H) \* E(X | H) + P(T) \* E(X | T)

= 1/2 \* (1 + E(T)) + 1/2 \* (1 + E(H))

= 1/2 \* (1 + 2) + 1/2 \* (1 + 2)

= **3**

### Solution 1c.

You flip heads twice.

**Method 1**:

TTT…H | TTT…H

Let X1 = # of flips till first H

Let X2 = # of flips till first H

This can be broken down into 2 random variables X1 and X2 where each one represents a Geometric random variable of flipping a heads.

By Linearity of Expectations, E(X1 + X2) = E(X1) + E(X2) = 2 + 2 = 4

Method 2:

E(X) = P(H) \* (1 + E(H)) + P(T) \* (1 + E(X))

= 1/2 (1 + 2) + 1/2 \* (1 + E(X))

E(X) = **4**

### Solution 1d.

You flip heads twice in a row

**Method 1**:

E(X) = P(H) \* E(X | H) + P(T) \* E(X | T)

> E(X | H) = P(H) \* E(X | HH) + P(T) \* E(X | HT) = 1/2 \* 2 + 1/2 \* (2 + E(X)) = 2 + E(X)/2

E(X) = 1/2 \* (2 + E(X)/2) + 1/2 \* (1 + E(X))

= **6**

**Method 2**:

E(X) = P(HH) \* E(X | HH) + P(HT) \* E(X | HT) + P(T) \* E(X | T)

= 1/4 \* 2 + 1/4 \* (2 + E(X)) + 1/2 \* (1 + E(X))

= **6**

### Solution 1e.

You flip heads followed immediately by tails

**Method 1**:

TTT…H | HHH…T

Let X1 = # of flips till first H

Let X2 = # of flips till first T

By Linearity of Expectations, E(X1 + X2) = E(X1) + E(X2) = 2 + 2 = **4**

**Method 2**:

Let X = flips till HT

Let Y = flips till first T given last is H

E(X) = P(H) \* (1+ E(Y)) + P(T) \* (1 + E(X))

= 1/2 \* (1+ 2) + 1/2 \* (1 + E(X))

= **4**

### Solution 2a.

2. *Reservoir sampling* is a method for choosing an item uniformly at random from an arbitrarily long stream of data. For example, the sequence of packets that pass through a router, or the sequence of IP addresses that access a given web page. Like all data stream algorithms, this algorithm must process each item in the stream quickly, using very little memory. Assume that Random(*k*) returns a uniform random value in {1*...k*}.

GetOneSample(stream *S*):

*l*← 0

while *S is not done* do

*x* ← next item in *S*

*l* ← *l* + 1

if *Random*(*l*) *= 1* then

*sample* ← *x* (\*)

return *sample*

At the end of the algorithm, the variable *l* stores the length of the input stream *S*; this number is *not* known in advance. If *S* is empty, the output of the algorithm is undefined. In the following, consider an arbitrary non-empty input stream *S* and let *n* denote the (unknown) length of *S*.

1

(a) Prove that the item returned by GetOneSample(*S*) is chosen uniformly at random from *S*.

Solution: base case when 1 item is there, prob of picking that item 1/1 =1

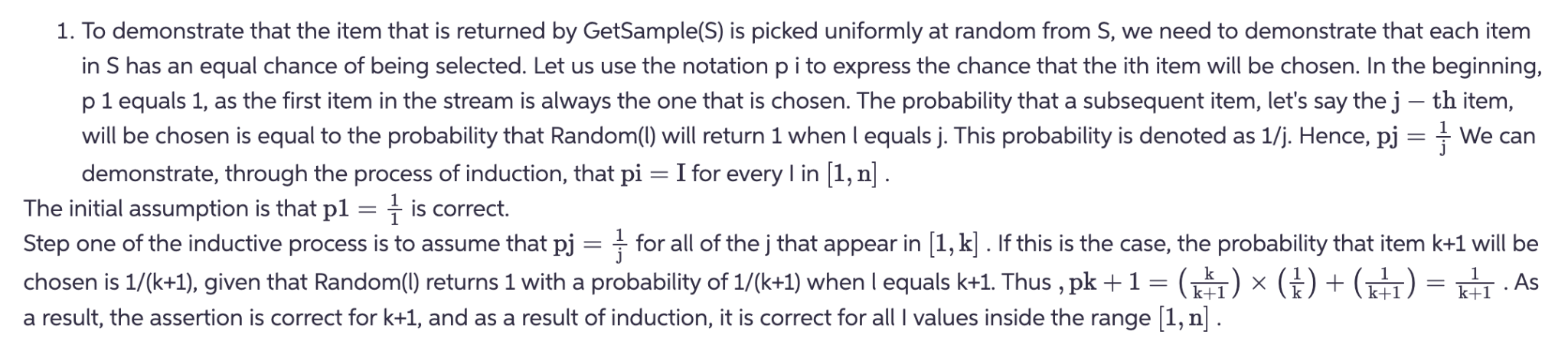
Induction : when k items are there prob of picking any item is 1/k.

Now lets say k+1 item arrives what is the probability that the first k items will also have an equally likely prob of being picked ?

Say we don’t pick k+1th item then prob of picking from prev k items is

Prob of not picking k+1th item \* prob that prev k item one was picked already

= k/(k+1) \* (1/k) = 1/k+1, hence each of the k prev items is also a prob of being picked.



### Solution 2b.

(b) What is the *exact* number of times that GetOneSample(*S*) executes line (\*)?

Solution: every time we can pick a 1 at random is the total times we will execute line \*. First time 1 can be picked with prob 1/1, second time with ½, this time with 1/3 and so on till 1/n

So sum of all these prob 1/1 + ½ + 1/3 + ¼ … 1/n = log(n)

The actual number of times that line 'O' will be executed is ln(n), where ln is the natural logarithm. This can be found by calling the GetSample(S) function. To see why this is the case, consider the fact that the chance of any specific item being chosen during the i-th iteration of the loop is equal to I As a result, the total of the harmonic series, which is written as ln(n) + O, represents the estimated number of iterations that will occur before an item is chosen (1).

### Solution 2c.

(c) What is the *exact* expected value of *l* when GetOneSample(*S*) executes line (\*) for the *last* time?

Solution: see pic below

### Solution 2d.

(d) What is the exact expected value of l when either GetOneSample(S) executes line (\*) for the second time or algorithm ends (whichever happens first)?

Solution: see pic below

