Parallel And Sequential Data Structures and Algorithms

Probability for Randomized Algorithms

Learning Objectives

- Recall **basic probability tools** to analyze random processes
- Identify the various different classes of randomized algorithms
- Apply the laws of **expectation** to solve randomized problems
- Learn how to apply expectation and **high probability bounds** to analyze work/span of randomized algorithms

Basic Probability

Definitions

Sample space (Ω): Set of possible outcomes of a “well defined experiment”. We’re doing discrete probability in this class, so Ω is finite or countably infinite.

Events: Arbitrary subsets of Ω (usually denoted by capital letters like A).

Probability measure (Pr): $\Pr : 2^{\Omega} \to \mathbb{R}$ with

1. $0 \leq \Pr(A) \leq 1$ for any event A
2. If $A \cap B = \varnothing$ then $\Pr(A) + \Pr(B) = \Pr(A \cup B)$
3. $\Pr(\Omega) = 1$
4. Since we’re being discrete just give each outcome x a Pr(x). And the probability of an event is just the sum of the Pr() of all outcomes in A.

Independence: A and B are independent if $\Pr(A \cap B) = \Pr(A) \cdot \Pr(B)$

Example

Random Distance Run: Assume we have two dice. Let D_1 be the value that the first die rolls, and D_2 be the value that the second die rolls. Our sample space is the set of possible rolls for the two dice, and D_1 and D_2 each take outcomes from the sample space and map them to real numbers in {1, 2, 3, 4, 5, 6}.

Here $\Omega = \{ (1,1), (1,2), \ldots (6,5), (6,6) \}$ $|\Omega| = 36$

The probability measure for two fair dice is that each of the 36 outcomes has equal probability, namely 1/36.

Example contd.

Random Distance Run: Assume we have two dice. Let D_1 be the value that the first die rolls, and D_2 be the value that the second die rolls. Our sample space is the set of possible rolls for the two dice, and D_1 and D_2 each take outcomes from the sample space and map them to real numbers in {1, 2, 3, 4, 5, 6}.

Let X = the event that D_1 = 1. X = {(1,1),(1,2),...,(1,6)} Pr(X) = 6/36 = 1/6.
Let Y = the event that D_2 = 3. Y = {(1,3),(2,3),...,(6,3)} Pr(Y) = 6/36 = 1/6.

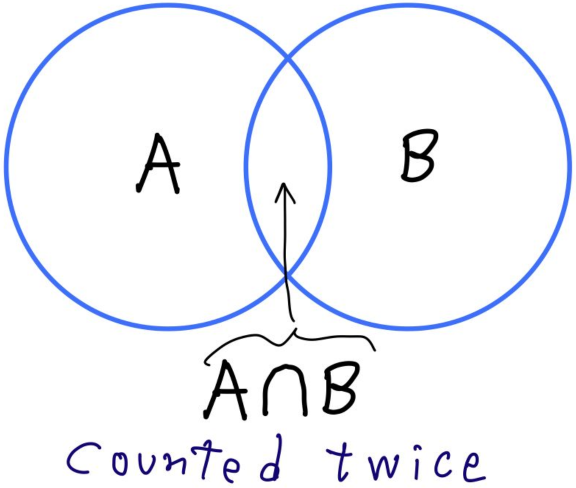
Pr(X∩Y) = Pr{(1,3)} = 1/36 = Pr(X) · Pr(Y) ... Thus X and Y are independent.

Venn Diagrams

We said that if A ∩ B = ∅ then Pr(A) + Pr(B) = Pr(A ∪ B). More generally, for arbitrary events A and B we have:

$$\Pr(A) + \Pr(B) = \Pr(A \cup B) + \Pr(A \cap B)$$

This can be visualized in the following Venn Diagram:

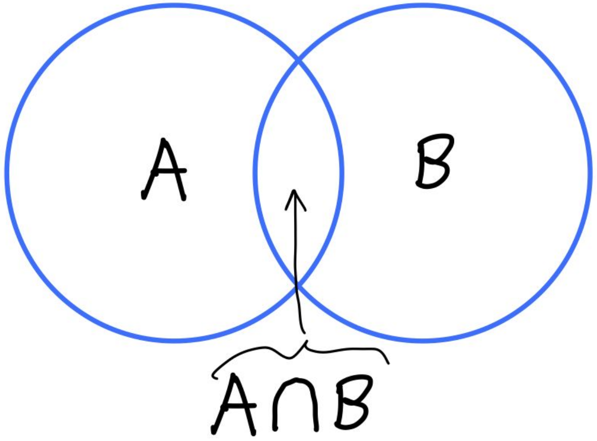


Union Bound

It follows that:

$$\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$$

This is known as the **Union Bound**.



Random Variables

A random variable X is a function X: Ω → ℝ

This endows each element of the RANGE of X with a probability.

Namely:

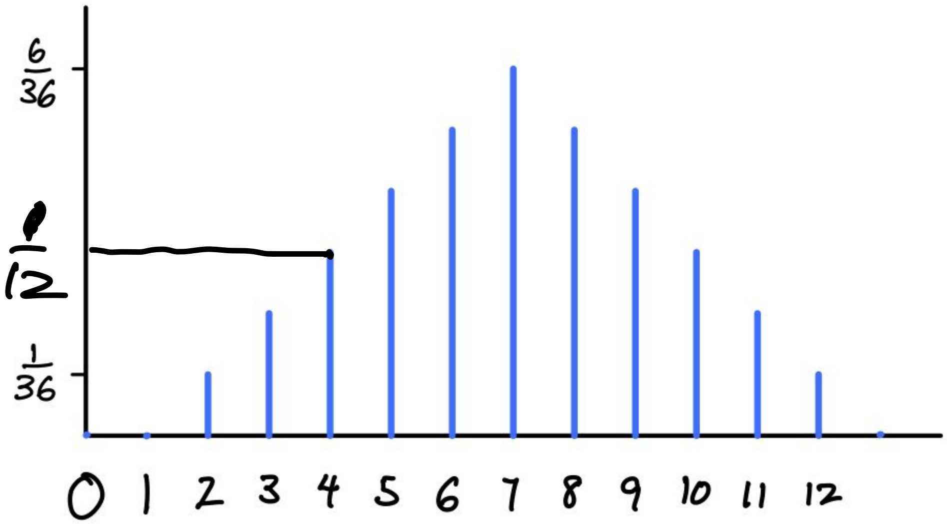
$$\Pr(\underbrace{X = x}_{\text{an event}}) = \sum_{\substack{e \in \Omega \\ X(e) = x}} \Pr(e)$$

Visualizing Random Variables

Consider a random variable X in our two dice example.
X: (a,b) ➡ a+b

What is Pr(X=4)?   It's Pr({1,3}) + Pr({2,2}) + Pr({3,1}) = 1/12

What about other values besides 4? The whole RV is shown below.



Sampling: An Alternative Intuition for RVs

A RV can be imagined as a randomized algorithm to **sample** the RV. This algorithm uses sources of randomness (e.g. coin flips or sampling other RVs) and outputs a number. In this fashion the algorithm produces a sample value of the RV.

It will often be useful to allow the sampling algorithm to depend on some parameter, e.g. an integer n.

Alternative Intuition for RVs

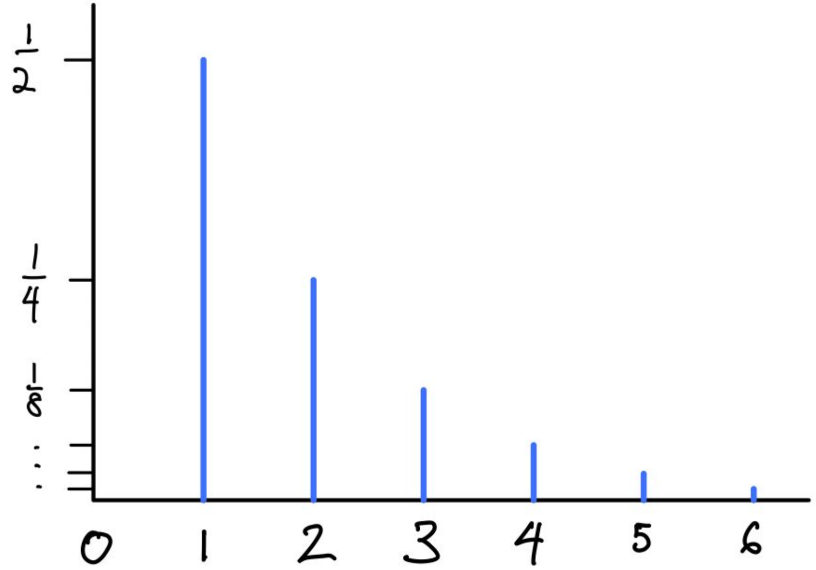
For example. Consider a RV which flips a fair coin until it comes up heads, and outputs the number of coin flips done.

What does the graphical representation of this RV look like?

Alternative Intuition for RVs

For example. Consider a RV which flips a fair coin until it comes up heads, and outputs the number of coin flips done.

What does the graphical representation of this RV look like?



More Definitions

Independence: Two random variables are independent if their events are independent. I.e.

$$\Pr(X=a, Y=b) = \Pr(X=a)\Pr(Y=b) \qquad \forall\, a,b$$

More Definitions

Expected Value: Weighted average over all possible outcomes of some random variable X. More formally,

$$\mathbb{E}[X] = \sum_{a \in \Omega} \Pr(a) \cdot X(a) = \sum_{x} x \cdot \Pr(X = x)$$

Indicator Random Variable: I_E represents the occurrence of event *E*. $I_E = 1$ if event *E* occurs or 0 otherwise. This is useful because $\mathbb{E}[I_E] = \Pr(E)$.

Combining RVs

Combining RVs: **X + Y** is a RV where $\Pr(X + Y = z) = \sum_{x,y} \Pr(X = x, Y = y), \forall x + y = z$

Similarly for other functional operators.

Combining RVs

Combining RVs: **X + Y** is a RV where $\Pr(X+Y=z)=\sum_{x,y}\Pr(X=x,Y=y),\forall x+y=z$

Similarly for other functional operators.

Alternatively, using the algorithmic sampling definition, **X+Y** is the following sampling algorithm for **X+Y**:

```
sample(X+Y):
    return (sample(X) + sample(Y))
```

Practice

Random Distance Run: Assume we have two dice. There are two random variables.

D_1 = roll of die #1
D_2 = roll of die #2

1. What is the expected value of D_1? 3.5
2. What is the expected value of D_1+D_2? 7

$E[D_1+D_2] = E[D_1]+E[D_2]$
$= 3.5+3.5 = 7$

Theorems

Linearity of Expectation: For two random variables X and Y, the expected value of their sum is the sum of their expected values. Formally,

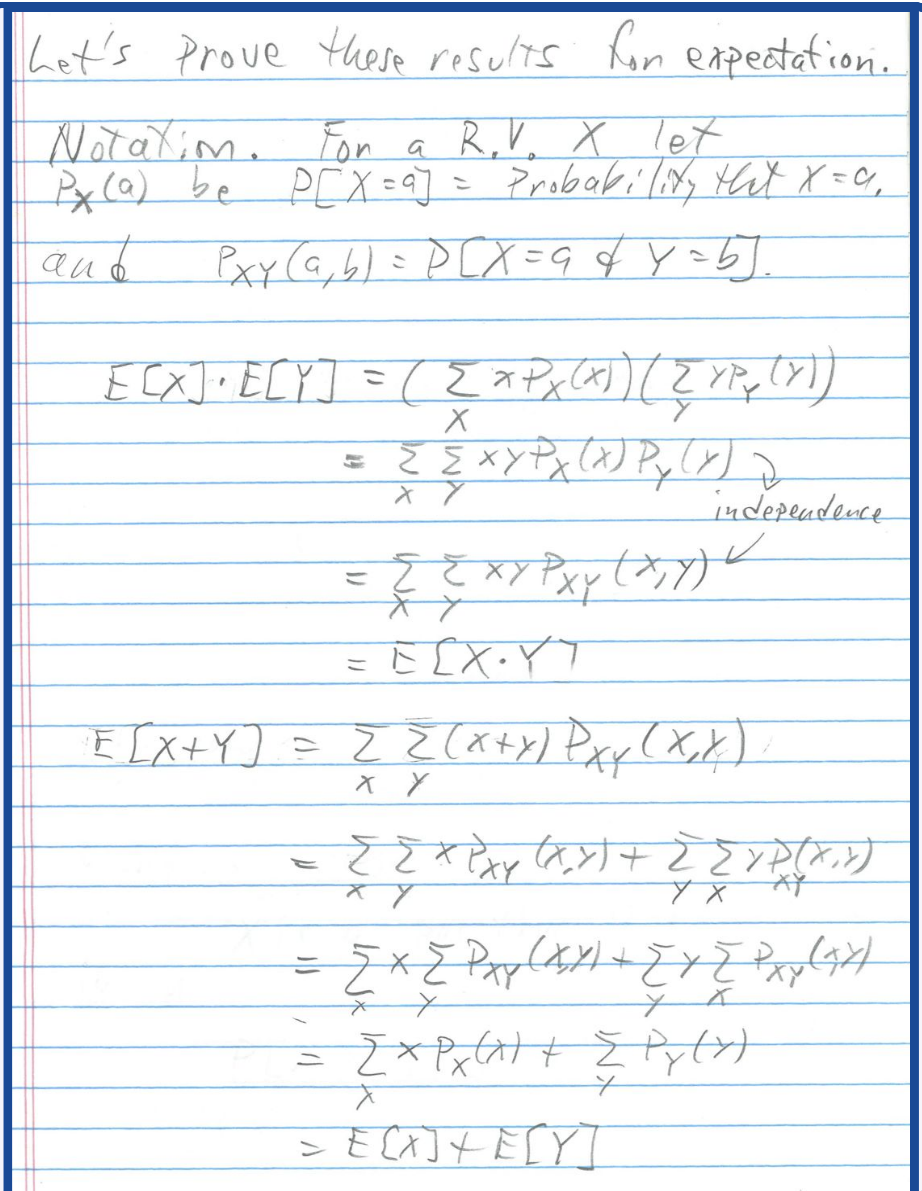
$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

Note that this is true even if X and Y are not independent!

Product of Expectation: For two ***independent*** random variables X and Y, the expected value of their product is the product of their expected values. Formally,

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$$

Proofs



Let's prove these results for expectation.

Notation. For a R.V. X let $P_X(a)$ be $P[X=a]$ = Probability that $X=a$,

and $P_{XY}(a,b) = P[X=a \,\&\, Y=b]$.

$$E[X]\cdot E[Y] = \Big(\sum_x x P_X(x)\Big)\Big(\sum_y y P_Y(y)\Big)$$

$$= \sum_x \sum_y xy P_X(x) P_Y(y)$$

independence

$$= \sum_x \sum_y xy P_{XY}(x,y)$$

$$= E[X\cdot Y]$$

$$E[X+Y] = \sum_x \sum_y (x+y) P_{XY}(x,y)$$

$$= \sum_x \sum_y x P_{XY}(x,y) + \sum_y \sum_x y P_{XY}(x,y)$$

$$= \sum_x x \sum_y P_{XY}(x,y) + \sum_y y \sum_x P_{XY}(x,y)$$

$$= \sum_x x P_X(x) + \sum_y P_Y(y)$$

$$= E[X] + E[Y]$$

Practice

Random Distance Run: Assume we have two dice. There are two random variables.

D_1 = roll of die #1
D_2 = roll of die #2

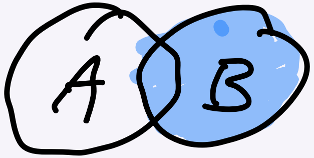
1. $\mathbb{E}[D_1 D_2]$ = $49/4 = 12\frac{1}{4}$
2. $\mathbb{E}[D_1 D_2$ where D_1 and D_2 are "entangled"] = $1+4+9+16+25+36)/6 = \frac{91}{6} = 15\frac{1}{6}$
3. $\mathbb{E}[D_1 + D_2$ where D_1 and D_2 are "entangled"] = 7
4. $\mathbb{E}[\max(D_1, D_2)]$ = $4\frac{7}{36}$

Final Definitions

Union Bound: For any two events A and B, the probability that either of them happens is upper bounded by their total probability, i.e.

$$\Pr(A \cup B) \leq \Pr(A) + \Pr(B)$$

Conditional Probability: The conditional probability of A given B, written as Pr(A | B), gives us the probability A happens *assuming* B has happened. It is defined to be as follows:



$$\Pr(A \mid B) = \frac{\Pr(A \cap B)}{\Pr(B)}$$

Notice how Pr(A | B) = Pr(A) when A and B are independent.

Tools

Harmonic Numbers

The n^th harmonic number H_n is defined as the sum of the reciprocals of the first n positive integers, i.e.

$$H_n = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} = \sum_{i=1}^{n} \frac{1}{i}$$

Theorem: The n^th harmonic number H_n, satisfies

$$\ln n < H_n < \ln n + 1$$

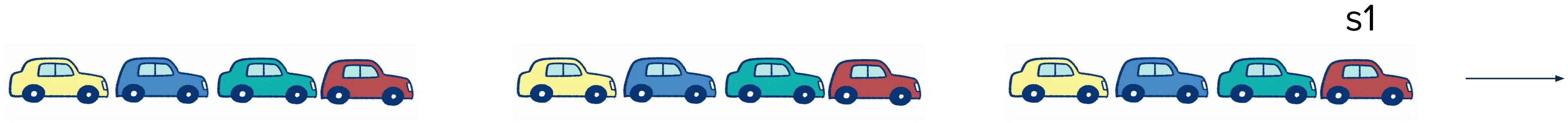
In other words, $H_n = \Theta(\log n)$, which we are happy to see in algorithms!

Exercise

Car Clusters: Suppose n cars are driving along a straight line, initially spaced so that none are touching. Each car is assigned a distinct maximum speed, chosen uniformly at random in [1,n]. Cars accelerate up to their maximum speed, but if a car reaches the one in front, it slows down to match that car's speed. (They're under cruise control.)

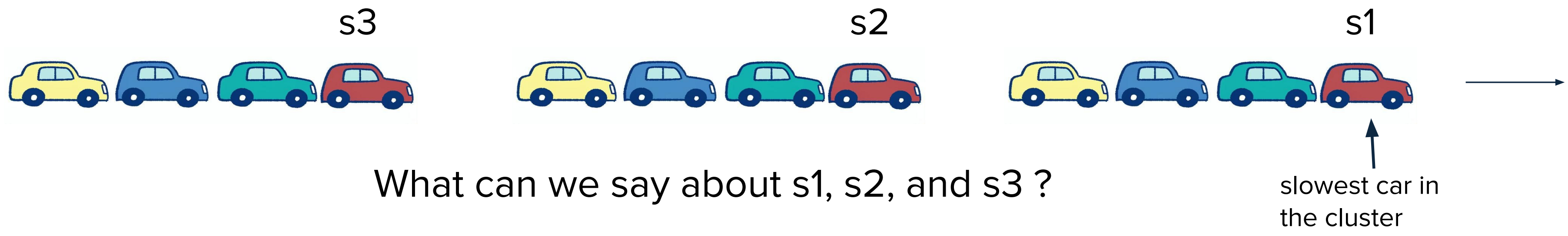
A *cluster* is a group of cars that end up driving together at the same speed. How many clusters should we expect?

Exercise



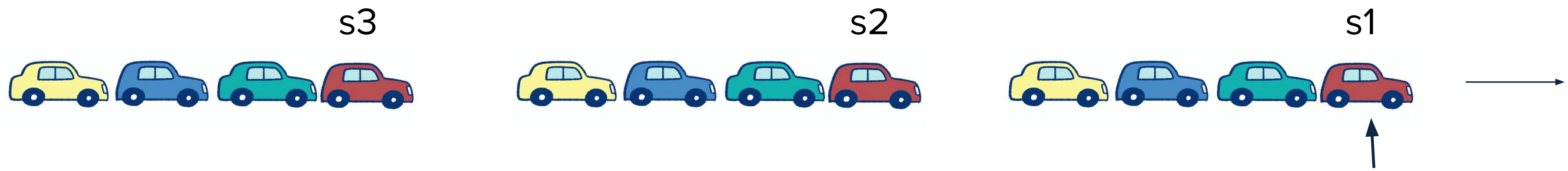
How does s1 compare to the rest in that cluster?

Exercise



What can we say about s1, s2, and s3 ?

Exercise



s3

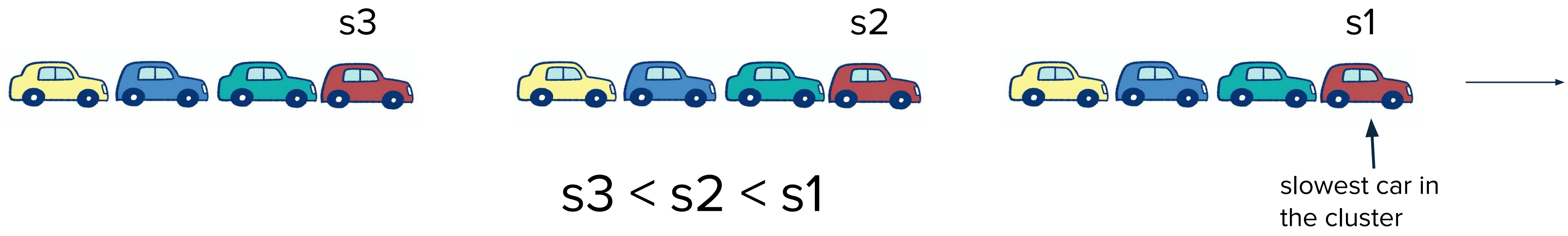
s2

s1

$s3 < s2 < s1$

slowest car in the cluster

Exercise



$s3 < s2 < s1$

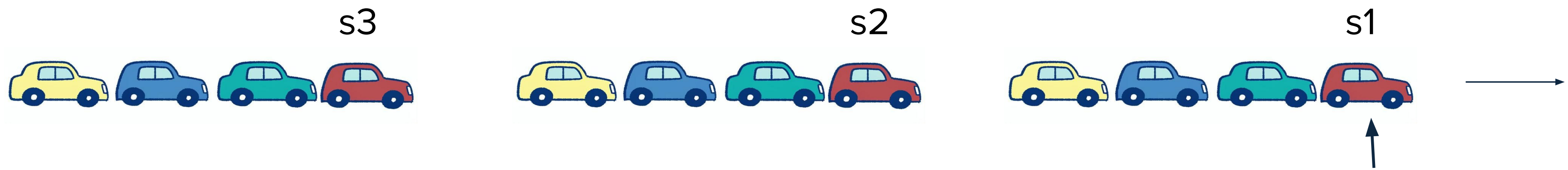
slowest car in the cluster

Therefore the front car in any cluster is slower than **all cars** to its right.

Therefore the number of clusters is the number of cars that are slower than all of the cars to its right.

The distinct speeds are assigned randomly. So what is the probability that the i’th car from the right is the slowest among all those to its right?

Exercise



$s3 < s2 < s1$

slowest car in the cluster

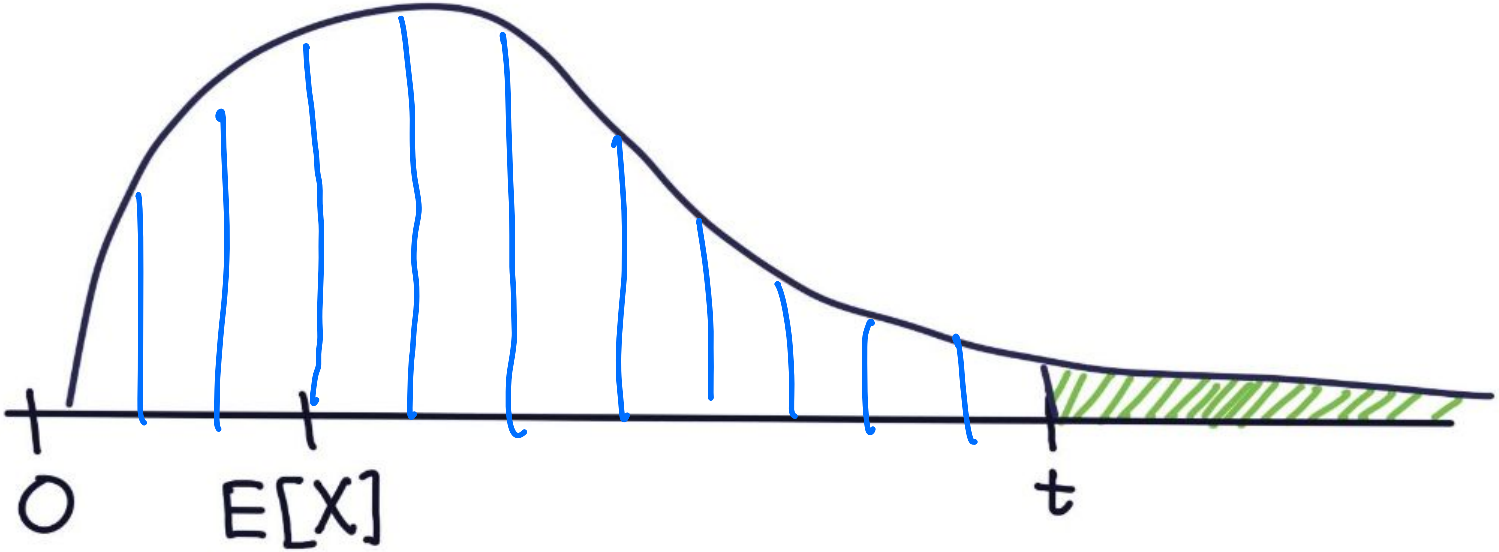
Therefore the front car in any cluster is slower than **all cars** to its right.

Therefore the number of clusters is the number of cars that are slower than all of the cars to its right.

The distinct speeds are assigned randomly. So what is the probability that the i’th car from the right is the slowest among all those to its right?

Answer: $\frac{1}{i}$ Make I_i this indicator variable. $\mathbb{E}[\#\text{clusters}] = \mathbb{E}[I_1+\ldots+I_n] = H_n$

Tail Bounds



Theorem

Markov's Inequality: For a random variable X which is always greater than or equal to 0, the following tail bound holds:

$$\Pr(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

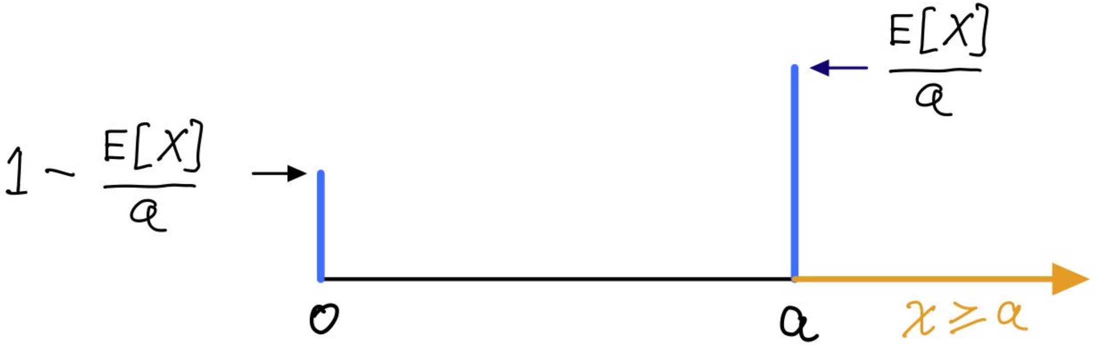
Intuitively, this means that if we have a probability distribution for X with some expected value $\mathbb{E}$[X], then Markov's inequality says we can't be above $\mathbb{E}$[X] very often.

Theorem

Markov's Inequality: For a random variable X which is always greater than or equal to 0, the following tail bound holds:

$$\Pr(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$$

Proof: Suppose X had this distribution:



This distribution has expectation $\mathbb{E}$[X]. If another distribution has more mass in the x ≥ a region, then that distribution's expectation is > $\mathbb{E}$[X]. ■

Analyzing Algorithms

Randomized Algorithms

Las Vegas Algorithm: A randomized algorithm whose cost bounds are a random variable, but will always return the correct result.

Monte Carlo Algorithm: A randomized algorithm whose cost bounds are deterministic, but it may fail or return incorrect answers. The context is usually a minimization problem, where the returned answer is a valid solution, but might not be of minimum cost. We know a probability $p>0$ that it will be optimum.

Randomized Algorithms

Las Vegas Algorithm: A randomized algorithm whose cost bounds are a random variable, but will always return the correct result.

Monte Carlo Algorithm: A randomized algorithm whose cost bounds are deterministic, but it may fail or return incorrect answers. The context is usually a minimization problem, where the returned answer is a valid solution, but might not be of minimum cost. We know a probability p>0 that it will be optimum.

Exercise: Can we turn a Las Vegas algorithm into a Monte Carlo algorithm?

Mnemonic for remembering this definition:

"In the Monte Hall problem, you make mistakes!"

Randomized Algorithms

Note that for Monte Carlo algorithms, you can run them repeatedly in order to **amplify** the probability that the correct solution has been found.

# runs	Probability that the best solution found is optimal
1	p
2	$1-(1-p)^2$
3	$1-(1-p)^3$
1/p	$1-(1-p)^{1/p} \simeq 1-\frac{1}{e}$
n/p	$1-\frac{1}{e^n}$

$(1-\frac{1}{n})^n \longrightarrow \frac{1}{e}$ as $n \to \infty$

(for large n)

QuickSort

Algorithm (QuickSort):

```
fun quicksort(S : sequence<T>) -> sequence<T>:
  if length(S) <= 1:
    return S
  p = uniform_random_element(S)
  L = filter(fn x => (x < p), S)
  R = filter(fn x => (x > p), S)
  M = filter(fn x => (x == p), S)
  sortedL, sortedR = parallel (quicksort(L), quicksort(R))
  return sortedL + M + sortedR
```

$T(n) = T(n-1) + n$

$= \Theta(n^2)$

is possible,

but improbable

Exercise: Is this a Las Vegas or Monte Carlo algorithm? Las Vegas

High Probability

Definition: We say that a random variable *W(n)* ≤ *f(n)* **with high probability** (w.h.p.) if there exists a constant n_0 such that for any integer value of *k* ≥ 1 and any $n \geq n_0$:

$$\Pr(W(n) \leq k \cdot f(n)) \geq 1 - \frac{1}{n^k}$$

We will also write *W(n)* ∈ *O(f(n))* w.h.p. if *W(n)* ≤ c · *f(n)* w.h.p. for some constant c.

We are essentially imposing a very strict tail bound on the random variable *W(n)*. An equivalent (sometimes more convenient) way to view it is

$$\Pr(W(n) > k \cdot f(n)) < \frac{1}{n^k}$$

Theorem

Max Preserves w.h.p.: Let $S(n)$ be a non-negative random variable, and let $T(n) = \max(S(n), \ldots, S(n))$, where there are n (not necessarily independent) copies of $S(n)$ in the max. If $S(n) \leq f(n)$ w.h.p. then $T(n) \leq 2f(n)$ w.h.p. It follows that if $S(n) \in O(f(n))$ w.h.p. then so is $T(n)$.

Theorem

Max Preserves w.h.p.: Let $S(n)$ be a non-negative random variable, and let $T(n) = \max(S(n), \dots, S(n))$, where there are n (not necessarily independent) copies of $S(n)$ in the max. If $S(n) \leq f(n)$ w.h.p. then $T(n) \leq 2f(n)$ w.h.p. It follows that if $S(n) \in O(f(n))$ w.h.p. then so is $T(n)$.

Proof:

$$\Pr(T(n) > (k+1)f(n)) \leq n \Pr(S(n) > (k+1)f(n)) \qquad \text{[Union Bound]}$$

But the RHS is at most $n/n^{k+1} = 1/n^k$ by virtue of $S(n) \leq f(n)$ w.h.p.
Also note that $2k \geq k+1$, so we can write:

$$\Pr(T(n) > k*2*f(n)) \leq \Pr(T(n) > (k+1)f(n)) \leq 1/n^k$$

which is the definition of $T(n) \leq 2f(n)$ w.h.p. ■

Skittles Game

Skittles Game: The Skittles game is played with a fair coin and a pile of *n* Skittles. It is a single-player game played in rounds. Initially there are *s* = *n* Skittles. Each round consists of the player flipping the coin once. If it comes up heads, then the player eats *s/2* (rounded up) of the Skittles and there are *s/2* (rounded down) remaining. If it comes up tails, the player proceeds to the next round without eating any Skittles. The game ends when there are no Skittles remaining. We are interested in the random variable *R(n)* which is the number of rounds the game lasts.

Wait a minute…

Skittles Game

∞

Theorem

Skittles Game Bound: The Skittles game will end in $5 \log_2 n$ rounds with high probability. In other words, $R(n) \in O(\log n)$ w.h.p.

Proof:

Theorem

Skittles Game Bound: The Skittles game will end in $5 \log_2 n$ rounds with high probability. In other words, $R(n) \in O(\log n)$ w.h.p.

Proof:

NEXT LECTURE!

Summary

- Learned about basic probability definitions and **random variables**
- Identified two different classes of random algorithms, our focus remains on **Las Vegas algorithms** and their runtime.
- Applied tail bound analysis in the form of **Markov's inequality** and **high probability bounds**
- To be continued: high probability bounds on the **Skittles Game**