CSE 250 Recitation

April 17 - 18: Expected Runtime

Expected Value

A **random variable** represents a quantity that is dependent on random occurrence

Example: Let **X** be the value rolled on a six-sided die

X is a random variable

- It's value can be 1,2,3,4,5 or 6
- It's value depends on a random event (rolling the die)

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Example: Let **X** be the value rolled on a six-sided die

Possible values of **X**: 1,2,3,4,5,6

Probability of each outcome: ⅙

E[X] = $\frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = 3.5$

Expected Value

The **expected value** of a random variable, **X**, is the average value of the possible outcomes, weighted by the probability of each outcome. Denoted **E[X]**.

Generally:

$$E[X] = \sum_i P_i \cdot X_i$$

Probability of the i^{th} outcome

value of the i^{th} outcome

Exercise

A deck of cards contains 52 cards. 4 aces, 4 of each number 2-10, and 12 face cards (4 jacks, 4 queens, 4 kings).

In Blackjack, number cards are worth their value (ie 2 is worth 2, 3 is worth 3, etc), face cards are worth 10, and aces (for simplicity) are worth 1.

If you draw a single card from a shuffled deck of cards, what is the expected value of that card?

$$E[X] = \sum_i P_i \cdot X_i$$

Exercise

Note: 4/52 = 1/13

If **X** is the value of the drawn card, then

E[X] = 1/13 · (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10) + 3/13 · 10 = 6.538

Linearity of Expectation

Expected value is a linear function; **E[X+Y] = E[X] + E[Y]** and **E[cX] = cE[X]**

Example: Let **X** and **Y** represent the value of two different rolls of a 6-sided die

The expected value of the sum of these rolls is **E[X + Y]**

We could compute this the long way by averaging over all 36 possible outcomes…but since expectation is linear:

Linearity of Expectation

Expected value is a linear function; **$E[X+Y] = E[X] + E[Y]$** and **$E[cX] = cE[X]$**

Example: Let **X** and **Y** represent the value of two different rolls of a 6-sided die

The expected value of the sum of these rolls is **$E[X + Y]$**

We could compute this the long way by averaging over all 36 possible outcomes…but since expectation is linear:

$E[X + Y] = E[X] + E[Y] = 3.5 + 3.5 = 7$

Discussion

How would we find the expected sum of rolling a d6 and a d20 together?

Discussion

How would we find the expected sum of rolling a d6 and a d20 together?

Since expectation is linear…compute the expected value of each and add them!
Let **X** be the d6, and **Y** be the d20

We already know **E[X] = 3.5**

E[Y] = 1/20 · (1 + 2 + 3 +...+ 19 + 20) = 10.5

E[X+Y] = 14

Discussion

How would we find th

Since expectation is l
Let **X** be the d6, and **Y**

We already know **E[X]**

E[Y] = 1/20 · (1 + 2 +

E[X+Y] = 14

See it in action:

Generate a bunch of die rolls: d6 d20

Compute the averages (for example in a spreadsheet)

The average of the d6 will be *close* to 3.5
The average of the d20 will be *close* to 10.5

The more rolls, the closer the averages will be to expectation

The average of the sums will always EXACTLY be the sum of the averages!

Expected Runtime Example #1

```
def mystery(data):
  if randInt() % 100 == 0:
    sum = 0
    for d in data:
      sum += data
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the growth function, T(n), representing the runtime of this function.

What are the unqualified bounds?

Expected Runtime Example #1

```
def mystery(data):
  if randInt() % 100 == 0:
    sum = 0
    for d in data:
      sum += data
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the growth function, T(n), representing the runtime of this function.

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

What are the unqualified bounds?

O(n), Ω(1)

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

The first outcome happens 1/100 times

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What is E[T(n)]?

$$T(n) = \begin{cases} n & \text{if X \% 100 == 0} \\ 1 & \text{otherwise} \end{cases}$$

Remember: $E[X] = \sum_i P_i \cdot X_i$

The second outcome happens 99/100 times

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What are the bounds of E[T(n)]?

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #1

Discussion: What are the bounds of E[T(n)]? **O(n)**

$$E[T(n)] = \frac{1}{100} \cdot n + \frac{99}{100} \cdot 1$$

Expected Runtime Example #2

```
def mystery(data):
  if randInt()%data.size()==0:
    sum = 0
    for d in data:
      sum += data
  else:
    sum = data[0] * data.size()
  return sum
```

Exercise:

Write out the runtime, T(n), and the expected runtime, E[T(n)] for this function.

What are the bounds on these growth functions?

Expected Runtime Example #2

```
def mystery(data):
  if randInt()%data.size()==0:
    sum = 0
    for d in data:
      sum += data
  else:
    sum = data[0] * data.size()
  return sum
```

$$T(n) = \begin{cases} n & \text{if X \% n == 0} \\ 1 & \text{otherwise} \end{cases} \in O(n), \Omega(1)$$

$$E[T(n)] = \frac{1}{n} \cdot n + \frac{n-1}{n} \cdot 1 \in O(1)$$

Expected Runtime Example #3

```
def mystery(data):
  idx = randInt() % data.size()
  val = data[idx]
  for i = 1..val:
    print("wow")
```

Discussion:

What is the worst-case runtime if:

- **Values in data range from 1-100?**
- **Values in data range from 1-n?**

What is the expected runtime if:

- **The expected value of data[x] is 20?**
- **The expected value of data[x] is n/2?**