CS 70 Midterm 2 Review

Fall 2018

October 27, 2018

Agenda

Agenda

- 6 Questions: You try problems, we present solutions.

Agenda

- 6 Questions: You try problems, we present solutions.
- Don't write up a complete solution to each question in the allotted time - but do think about an approach.

Agenda

- 6 Questions: You try problems, we present solutions.
- Don't write up a complete solution to each question in the allotted time - but do think about an approach.
- Questions: After we present solutions.

RSA for Midterm Scores

Alice wants to tell Bob her midterm score, m, which is an integer between 0 and 100 inclusive. She wants to tell Bob over an insecure channel that Eve can listen in on, but Alice does not want Eve to know Alice's midterm score.

1. Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?
2. Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m?

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

- Adding and multiplying n bit numbers is easy (think $n \approx 512$)

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

- Adding and multiplying n bit numbers is easy (think $n \approx 512$)
- Computing $x^e \pmod{N}$ is easy (via repeated squaring)

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

- Adding and multiplying n bit numbers is easy (think $n \approx 512$)
- Computing $x^e \pmod N$ is easy (via repeated squaring)
- Computing $x^{-1} \pmod N$, and $\gcd(x, y)$ is easy, when x and y are ≈ 512 bits.

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

- Adding and multiplying n bit numbers is easy (think $n \approx 512$)
- Computing $x^e \pmod N$ is easy (via repeated squaring)
- Computing $x^{-1} \pmod N$, and $\gcd(x, y)$ is easy, when x and y are ≈ 512 bits.
- Determining if a number N is prime is easy (the reason for this is beyond scope of this course). Factoring a number N is believed to be *hard*.

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

Let's review what computations are tractable, and which are not.

- Adding and multiplying n bit numbers is easy (think $n \approx 512$)
- Computing $x^e \pmod N$ is easy (via repeated squaring)
- Computing $x^{-1} \pmod N$, and $\gcd(x, y)$ is easy, when x and y are ≈ 512 bits.
- Determining if a number N is prime is easy (the reason for this is beyond scope of this course). Factoring a number N is believed to be *hard*.
- Taking the e-th root in a mod is *hard* i.e. given $x^e \pmod N$, finding x is hard.

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

- Alice's m (plaintext) only has 101 possibilities! She sends $m^e \pmod{N}$ to Bob.

Solution

Bob announces his public key $(N = pq, e)$, where N is large (512 bits). Alice encrypts her message using RSA. Eve sees the encrypted message, and figures out what Alice's midterm score is. How did she do it?

Solution:

- Alice's m (plaintext) only has 101 possibilities! She sends $m^e \pmod{N}$ to Bob.
- Eve can just compute $0^e, 1^e, 2^e, \ldots, 100^e \pmod{N}$, check which one matches, and then she has successfully figured out Alice's score!

Solution

Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m?

Solution:

- Let's first figure out what exactly Alice sends to Bob.

Solution

Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m?

Solution:

- Let's first figure out what exactly Alice sends to Bob.
- She sends $(rm)^e \pmod N$ and $r^e \pmod N$ to Bob.

Solution

Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m?

Solution:

- Let's first figure out what exactly Alice sends to Bob.
- She sends $(rm)^e \pmod N$ and $r^e \pmod N$ to Bob.
- Eve doesn't even care about what r is. Eve can compute inverses efficiently using the extended GCD algorithm. Thus, she can find $r^{-e} \pmod N$. Once she does that, she can multiply that with $(rm)^e \pmod N$ to get back $m^e \pmod N$.

Solution

Alice decides to be a bit more elaborate. She picks a random number r that is 256 bits long, so that it is too hard to guess. She encrypts that and sends it to Bob, and also computes rm, encrypts that, and sends it to Bob. Eve is aware of what Alice did, but does not know the value of r. How can she figure out m?

Solution:

- Let's first figure out what exactly Alice sends to Bob.
- She sends $(rm)^e \pmod N$ and $r^e \pmod N$ to Bob.
- Eve doesn't even care about what r is. Eve can compute inverses efficiently using the extended GCD algorithm. Thus, she can find $r^{-e} \pmod N$. Once she does that, she can multiply that with $(rm)^e \pmod N$ to get back $m^e \pmod N$.
- Carry out the same attack as in part 1!

Hilbert's Paradox of the Grand Hotel

Consider a magical hotel with a countably infinite number of rooms numbered according to the natural numbers where all the rooms are currently occupied. Assume guests don't mind being moved out of their current room as long as they can get to their new room in a finite amount of time.

1. Suppose one new guest arrived in their car, how would you shuffle guests around to accommodate them? What if k guests arrived, where k is a constant positive integer?
2. Suppose a countably infinite number of guests arrived in an infinite length bus with seat numbers according to the natural numbers, how would you accommodate them?
3. Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution

Suppose one new guest arrived in their car, how would you shuffle guests around to accommodate them? What if k guests arrived, where k is a constant positive integer?

Solution: Our hotel has a (countably) infinite number of rooms!

Solution

Suppose one new guest arrived in their car, how would you shuffle guests around to accommodate them? What if k guests arrived, where k is a constant positive integer?

Solution: Our hotel has a (countably) infinite number of rooms!

- Shift every guest down the hall by k rooms (Guest in Room $i \to$ Room $i + k$).

Solution

Suppose one new guest arrived in their car, how would you shuffle guests around to accommodate them? What if k guests arrived, where k is a constant positive integer?

Solution: Our hotel has a (countably) infinite number of rooms!

- Shift every guest down the hall by k rooms (Guest in Room $i \to$ Room $i + k$).
- Put k new guests into the vacant first k rooms.

Solution

Suppose one new guest arrived in their car, how would you shuffle guests around to accommodate them? What if k guests arrived, where k is a constant positive integer?

Solution: Our hotel has a (countably) infinite number of rooms!

- Shift every guest down the hall by k rooms (Guest in Room $i \to$ Room $i + k$).
- Put k new guests into the vacant first k rooms.

Interpret: Since there is a bijection from the rooms in the hotel (the natural numbers) to the guests in the hotel after k more arrive (the naturals, plus a finite k), those two sets have the same cardinality - same "order of infinity".

Suppose a countably infinite number of guests arrived in an infinite length bus with seat numbers according to the natural numbers, how would you accommodate them?

Solution:
Our hotel has a (countably) infinite number of rooms!

Suppose a countably infinite number of guests arrived in an infinite length bus with seat numbers according to the natural numbers, how would you accommodate them?

Solution:
Our hotel has a (countably) infinite number of rooms!

- Shift every guest twice the distance down the hall as before (Guest in Room $i \to$ Room $2i$) so that only even-numbered rooms are occupied

Suppose a countably infinite number of guests arrived in an infinite length bus with seat numbers according to the natural numbers, how would you accommodate them?

Solution:
Our hotel has a (countably) infinite number of rooms!

- Shift every guest twice the distance down the hall as before (Guest in Room $i \to$ Room $2i$) so that only even-numbered rooms are occupied
- Put new guests in the odd-numbered rooms (Guest on bus seat number $j \to$ Room $2j + 1$)

Suppose a countably infinite number of guests arrived in an infinite length bus with seat numbers according to the natural numbers, how would you accommodate them?

Solution:
Our hotel has a (countably) infinite number of rooms!

- Shift every guest twice the distance down the hall as before (Guest in Room $i \to$ Room $2i$) so that only even-numbered rooms are occupied
- Put new guests in the odd-numbered rooms (Guest on bus seat number $j \to$ Room $2j + 1$)

Interpret: Since there is a bijection from the rooms in the hotel (the natural numbers) to the guests in the hotel after another infinite busload arrive (two times the size of the naturals), those two sets must still be on the same "order of infinity".

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

- Number the buses by their order of arrival $j = 1, 2, \ldots$ Consider the existing guests in the hotel as $j = 0$.

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

- Number the buses by their order of arrival $j = 1, 2, \ldots$ Consider the existing guests in the hotel as $j = 0$.
- Each guest in the bus has a seat number $i = 0, 1, 2, \ldots$ (The seat number for those already in the hotel is just their existing room number).

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

- Number the buses by their order of arrival $j = 1, 2, \ldots$ Consider the existing guests in the hotel as $j = 0$.
- Each guest in the bus has a seat number $i = 0, 1, 2, \ldots$ (The seat number for those already in the hotel is just their existing room number).

Rephrasing the question: Is there a bijection from the rooms in the hotel (the natural numbers) to the set of points in 2D (i, j) where $i = 0, 1, 2, \ldots$ and $j = 1, 2, \ldots$?

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

- Number the buses by their order of arrival $j = 1, 2, \ldots$ Consider the existing guests in the hotel as $j = 0$.
- Each guest in the bus has a seat number $i = 0, 1, 2, \ldots$ (The seat number for those already in the hotel is just their existing room number).

Rephrasing the question: Is there a bijection from the rooms in the hotel (the natural numbers) to the set of points in 2D (i, j) where $i = 0, 1, 2, \ldots$ and $j = 1, 2, \ldots$?
Yes! Same as enumerating the positive rationals. $\mathbb{N}$ has the same cardinality as $\mathbb{N}^2$!

Suppose a countably infinite number of infinite length buses arrive carrying countably infinite guests each, how would you accommodate them?

Solution:

- Number the buses by their order of arrival $j = 1, 2, \ldots$ Consider the existing guests in the hotel as $j = 0$.
- Each guest in the bus has a seat number $i = 0, 1, 2, \ldots$ (The seat number for those already in the hotel is just their existing room number).

Rephrasing the question: Is there a bijection from the rooms in the hotel (the natural numbers) to the set of points in 2D (i, j) where $i = 0, 1, 2, \ldots$ and $j = 1, 2, \ldots$?
Yes! Same as enumerating the positive rationals. $\mathbb{N}$ has the same cardinality as $\mathbb{N}^2$!
E.g. current guests go to even rooms. New guests into odd rooms in same order as $\mathbb{N}^2$ enumeration from lecture/notes.

Computability

1. The problem of determining whether a program halts in time 2^{n^2} on an input of size n is undecidable. (True or False.)
2. There is no computer program `DEAD` which takes a program P, an input x, and a line number n, and determines whether the nth line of code is executed when the program P is run on the input x. (True or False.)

Solution

The problem of determining whether a program halts in time 2^{n^2} on an input of size n is undecidable. (True or False.)

Solution:
False.

Solution

The problem of determining whether a program halts in time 2^{n^2} on an input of size n is undecidable. (True or False.)

Solution:
False. You can simulate a program for 2^{n^2} steps and see if it halts. The concept is that you can run a program for any fixed *finite* amount of time to see what it does. The problem of undecidability comes in when you don't have a bound on the time.

Solution

There is no computer program `DEAD` *which takes a program* P*, an input* x*, and a line number* n*, and determines whether the* n*th line of code is executed when the program* P *is run on the input* x*. (True or False.)*

Solution: True.
We implement `HALT` which takes a program P and an input x using `DEAD` as follows.

Solution

There is no computer program `DEAD` *which takes a program* P*, an input* x*, and a line number* n*, and determines whether the* n*th line of code is executed when the program* P *is run on the input* x*. (True or False.)*

Solution: True.
We implement `HALT` which takes a program P and an input x using `DEAD` as follows.
Modify P so that each exit or return statement jumps to an appended return statement at line $n+1$. Call the resulting program P'.

Solution

There is no computer program `DEAD` *which takes a program* P*, an input* x*, and a line number* n*, and determines whether the* n*th line of code is executed when the program* P *is run on the input* x*. (True or False.)*

Solution: True.
We implement `HALT` which takes a program P and an input x using `DEAD` as follows.
Modify P so that each exit or return statement jumps to an appended return statement at line $n+1$. Call the resulting program P'. We then hand P' to `DEAD` along with the input x and line $n+1$.

Solution

There is no computer program `DEAD` *which takes a program* P*, an input* x*, and a line number* n*, and determines whether the* n*th line of code is executed when the program* P *is run on the input* x*. (True or False.)*

Solution: True.
We implement `HALT` which takes a program P and an input x using `DEAD` as follows.
Modify P so that each exit or return statement jumps to an appended return statement at line $n+1$. Call the resulting program P'. We then hand P' to `DEAD` along with the input x and line $n+1$. If the original program halts than `DEAD` would return true, and if not `DEAD` would return false.

Solution

There is no computer program `DEAD` *which takes a program* P*, an input* x*, and a line number* n*, and determines whether the* n*th line of code is executed when the program* P *is run on the input* x*. (True or False.)*

Solution: True.
We implement `HALT` which takes a program P and an input x using `DEAD` as follows.
Modify P so that each exit or return statement jumps to an appended return statement at line $n+1$. Call the resulting program P'. We then hand P' to `DEAD` along with the input x and line $n+1$. If the original program halts than `DEAD` would return true, and if not `DEAD` would return false.
This contradicts the fact that the program `HALT` does not exist, so `DEAD` does not exist.

Solution

```
def HALT(P,x):
    def P'(x):
        run P(x)
        return x
    return DEAD(P',x,len(P'))
```

Tip

There are two ways to show undecidability.

Tip

There are two ways to show undecidability.

Use your program as a subroutine to solve a problem we know is undecidable or to do a "diagonalization" proof like we did for `HALT`.

Tip

There are two ways to show undecidability.

Use your program as a subroutine to solve a problem we know is undecidable or to do a "diagonalization" proof like we did for `HALT`.

The former is natural for computer programmers and flows from the fact that you are given P as text! Therefore you can look at it and make modifications. This is what the solution above does.

Finicky Bins

If a bin has at least 5 balls in a bin, the 5 balls will fall out and not be counted (e.g., 6 balls in a bin is the same as 1). Compute the number of ways to distribute 7 indistinguishable balls among 4 bins.

Solution

Consider a normal bin, in which balls do not disappear. With stars and bars we see there are $\binom{10}{3}$ ways to distribute the balls.

Solution

Consider a normal bin, in which balls do not disappear. With stars and bars we see there are $\binom{10}{3}$ ways to distribute the balls.

What if one bin has ≥ 5 balls?

Solution

There are 4 ways to choose which bin has ≥ 5 balls, and once we throw 5 balls into that bin, we are left to distribute 2 balls among 4 bins in $\binom{5}{3}$ ways.

Solution

There are 4 ways to choose which bin has ≥ 5 balls, and once we throw 5 balls into that bin, we are left to distribute 2 balls among 4 bins in $\binom{5}{3}$ ways.

Now, for the bin in which balls *do* disappear.

Solution

There are 4 ways to choose which bin has ≥ 5 balls, and once we throw 5 balls into that bin, we are left to distribute 2 balls among 4 bins in $\binom{5}{3}$ ways.

Now, for the bin in which balls *do* disappear.

There are $\binom{10}{3} - 4\binom{5}{3}$ ways to distribute the balls such that no balls disappear.

Solution

There are 4 ways to choose which bin has ≥ 5 balls, and once we throw 5 balls into that bin, we are left to distribute 2 balls among 4 bins in $\binom{5}{3}$ ways.

Now, for the bin in which balls *do* disappear.

There are $\binom{10}{3} - 4\binom{5}{3}$ ways to distribute the balls such that no balls disappear. There are $4\binom{5}{3}$ ways to distribute the balls such that 5 balls disappear, except that no matter where the disappearing balls are, there the resulting distribution of balls is the same. Therefore, we have to divide by 4 and we obtain $\binom{5}{3}$ ways to distribute the balls such that 5 balls disappear.

Solution

There are 4 ways to choose which bin has ≥ 5 balls, and once we throw 5 balls into that bin, we are left to distribute 2 balls among 4 bins in $\binom{5}{3}$ ways.

Now, for the bin in which balls *do* disappear.

There are $\binom{10}{3} - 4\binom{5}{3}$ ways to distribute the balls such that no balls disappear. There are $4\binom{5}{3}$ ways to distribute the balls such that 5 balls disappear, except that no matter where the disappearing balls are, there the resulting distribution of balls is the same. Therefore, we have to divide by 4 and we obtain $\binom{5}{3}$ ways to distribute the balls such that 5 balls disappear.

In total, we have

$$\binom{10}{3} - 4\binom{5}{3} + \binom{5}{3} = \binom{10}{3} - 3\binom{5}{3}.$$

Crazy Balls and Bins

Imagine you had 5 distinct bins and randomly threw 7 identical balls into the bins with uniform probability.

1. What is the probability that the first bin has exactly 3 balls in it?
2. What is the probability that the first bin has at least 3 balls in it?
3. What is the probability that at least one bin has exactly 3 balls in it?
4. What is the probability that at least one bin has at least 3 balls in it?

Crazy Balls and Bins - Solution

(exactly 3)

- WLG: Pr[ball 1,2,3 in first bin] $= \left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^4$.
- Because there could be any three balls

$$\binom{7}{3} \left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^4.$$

Crazy Balls and Bins - Solution

(exactly 3)

- WLG: Pr[ball 1,2,3 in first bin] $= \left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^4$.
- Because there could be any three balls

$$\binom{7}{3} \left(\frac{1}{5}\right)^3 \left(\frac{4}{5}\right)^4.$$

(at least 3)

- **at least 3** $\Longrightarrow$ summation from 3 to 7.
-

$$\sum_{k=3}^{7} \binom{7}{k} \left(\frac{1}{5}\right)^k \left(\frac{4}{5}\right)^{7-k}.$$

Crazy Balls and Bins - Solution

3.

- $$\sum_{i=1}^{5} \Pr[\text{bin i has exactly 3}] = 5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4$$
- overlap?

Crazy Balls and Bins - Solution

3.

- $$\sum_{i=1}^{5} \Pr[\text{bin i has exactly 3}] = 5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4$$
- overlap? yes!

Crazy Balls and Bins - Solution

3.

- $$\sum_{i=1}^{5} \Pr[\text{bin i has exactly 3}] = 5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4$$
- overlap? yes! $3+3+1 = 7$

Crazy Balls and Bins - Solution

3.

- $$\sum_{i=1}^{5} \Pr[\text{bin i has exactly 3}] = 5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4$$
- overlap? yes! $3+3+1 = 7$
- over-counting by

$$\binom{5}{2}\cdot\binom{7}{3}\binom{4}{3}\left(\frac{1}{5}\right)^6\frac{3}{5} = \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}$$

- Therefore our answer is

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}.$$

Crazy Balls and Bins - Solution

4.

- Pr[at least one bin has exactly 3] =

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}.$$

- Pr[at least one bin has exactly 4,5,6,7] ?
- $$5\sum_{k=4}^{7}\binom{7}{k}\left(\frac{1}{5}\right)^k\left(\frac{4}{5}\right)^{7-k}$$
- over-counting?

Crazy Balls and Bins - Solution

4.

- Pr[at least one bin has exactly 3] =

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}.$$

- Pr[at least one bin has exactly 4,5,6,7] ?
- $$5\sum_{k=4}^{7}\binom{7}{k}\left(\frac{1}{5}\right)^k\left(\frac{4}{5}\right)^{7-k}$$
- over-counting? Yes!

Crazy Balls and Bins - Solution

4.

- Pr[at least one bin has exactly 3] =

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}.$$

- Pr[at least one bin has exactly 4,5,6,7] ?
- $$5\sum_{k=4}^{7}\binom{7}{k}\left(\frac{1}{5}\right)^k\left(\frac{4}{5}\right)^{7-k}$$
- over-counting? Yes! $3+4=7$ or $4+3=7$

Crazy Balls and Bins - Solution

4.

- Pr[at least one bin has exactly 3] =

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}.$$

- Pr[at least one bin has exactly 4,5,6,7] ?
- $$5\sum_{k=4}^{7}\binom{7}{k}\left(\frac{1}{5}\right)^k\left(\frac{4}{5}\right)^{7-k}$$
- over-counting? Yes! $3+4=7$ or $4+3=7$
- over-counting by

$$2\cdot\binom{5}{2}\binom{7}{3}\left(\frac{1}{5}\right)^7$$

Crazy Balls and Bins - Solution

4.

solution:

$$5\binom{7}{3}\left(\frac{1}{5}\right)^3\left(\frac{4}{5}\right)^4 - \binom{5}{2}\frac{7!}{3!3!}\left(\frac{1}{5}\right)^6\frac{3}{5}$$
$$+5\sum_{k=4}^{7}\binom{7}{k}\left(\frac{1}{5}\right)^k\left(\frac{4}{5}\right)^{7-k} - 2\cdot\binom{5}{2}\binom{7}{3}\left(\frac{1}{5}\right)^7$$

Binomial Theorem

The binomial theorem states the following:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Prove this theorem using a combinatorial proof. You may assume that x and y are positive integers.

Binomial Theorem - Solution

combinatorial proof: different ways to do the same *task*.

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

observation:
LHS:

- only multiplications $\Longrightarrow$ separating the task into many steps

Binomial Theorem - Solution

combinatorial proof: different ways to do the same *task*.

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

observation:
LHS:

- only multiplications $\Longrightarrow$ separating the task into many steps

RHS:

- Summation $\Longrightarrow$ dividing the task into many *independent situations* that all together cover all possible situations.

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- n? x? y?

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- n? x? y?
- LHS:

$$(x+y)^n = (x+y)(x+y)(x+y)...(x+y) \implies$$

Number of ways of throwing n balls to (x+y) bins.

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- n? x? y?
- LHS:

$$(x+y)^n = (x+y)(x+y)(x+y)...(x+y) \implies$$

Number of ways of throwing n balls to (x+y) bins.

- check if this makes sense for RHS (if not, modify our "story")

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- $n \equiv$ number of balls, $(x+y) \equiv$ number of bins, task: counting the number of ways to throw n balls into $(x+y)$ bins

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- $n \equiv$ number of balls, $(x+y) \equiv$ number of bins, task: counting the number of ways to throw n balls into $(x+y)$ bins
- x? y?

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Thinking Process:

- $n \equiv$ number of balls, $(x+y) \equiv$ number of bins, task: counting the number of ways to throw n balls into $(x+y)$ bins
- x? y?
- RHS: each term is saying: choose k balls from n balls. Then throw the k balls to y particular bins, and the rest to the remaining x bins.

Binomial Theorem - Solution

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Formulate our story:

- $n \equiv$ number of balls,
- $(x+y) \equiv$ number of bins
- $x \equiv$ number of red bins
- $y \equiv$ number blue bins
- task: counting the number of ways to throw n balls into $(x+y)$ bins in which x of them are red, and y of them are blue.
- LHS: brutal and straight forward way
- RHS: (0 ball in blue bin + 1 ball in blue bin + 2 balls in blue bin + ... + all balls in blue bin)

Box of Marbles

You are given two boxes: one of them containing 900 red marbles and 100 blue marbles, the other one contains 500 red marbles and 500 blue marbles.

1. If we pick one of the boxes randomly, and pick a marble what is the probability that it is blue?
2. If we see that the marble is blue, what is the probability that it is chosen from box 1?
3. Suppose we pick one marble from box 1 and without looking at its color we put it aside. Then we pick another marble from box 1. What is the probability that the second marble is blue?

Solution

Solution

1. Define appropriate events.

Solution

1. Define appropriate events. Figure out what event you're trying to calculate the probability of - often that's half the problem!

Solution

1. Define appropriate events. Figure out what event you're trying to calculate the probability of - often that's half the problem!
   Let B = event that the picked marble is blue,
   A_1 = event that the marble is picked from box 1,
   A_2 = event that the marble is picked from box 2

Solution

1. Define appropriate events. Figure out what event you're trying to calculate the probability of - often that's half the problem!
   Let B = event that the picked marble is blue,
   A_1 = event that the marble is picked from box 1,
   A_2 = event that the marble is picked from box 2
   We want to calculate $\Pr(B)$. By total probability,

Solution

1. Define appropriate events. Figure out what event you're trying to calculate the probability of - often that's half the problem!
   Let B = event that the picked marble is blue,
   A_1 = event that the marble is picked from box 1,
   A_2 = event that the marble is picked from box 2
   We want to calculate $\Pr(B)$. By total probability,

$$\begin{aligned}\Pr(B) &= \Pr(B \mid A_1)P(A_1) + \Pr(B \mid A_2)\Pr(A_2) \\ &= 0.5 \times 0.1 + 0.5 \times 0.5 \\ &= 0.3\end{aligned}$$

Solution (cont.)

2. What probability do we want to calculate?

Solution (cont.)

2. What probability do we want to calculate? $\Pr(A_1 \mid B)$.

Solution (cont.)

2. What probability do we want to calculate? $\Pr(A_1 \mid B)$.
   By Bayes rule,

$$\begin{aligned}
\Pr(A_1 \mid B) &= \frac{\Pr(A_1 \cap B)}{\Pr(B)} \\
&= \frac{\Pr(B \mid A_1)\Pr(A_1)}{\Pr(B \mid A_1)\Pr(A_1) + \Pr(B \mid A_2)\Pr(A_2)} \\
&= \frac{0.1 \times 0.5}{0.5 \times 0.1 + 0.5 \times 0.5} \\
&= \frac{1}{6}
\end{aligned}$$

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble. We want to find $\Pr(B_2)$.

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.
   We want to find $\Pr(B_2)$.
   By total probability,

$$\begin{aligned}\Pr(B_2) &= \Pr(B_2 \mid B_1)\Pr(B_1) + \Pr(B_2 \mid R_1)\Pr(R_1) \\ &= \frac{99}{999} \times 0.1 + \frac{100}{999} \times 0.9 \\ &= 0.1\end{aligned}$$

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.
   We want to find $\Pr(B_2)$.
   By total probability,

$$\begin{aligned}\Pr(B_2) &= \Pr(B_2 \mid B_1)\Pr(B_1) + \Pr(B_2 \mid R_1)\Pr(R_1) \\ &= \frac{99}{999} \times 0.1 + \frac{100}{999} \times 0.9 \\ &= 0.1\end{aligned}$$

   But what if we had asked for the probability the nth marble is blue?

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.
   We want to find $\Pr(B_2)$.
   By total probability,

$$\begin{aligned}\Pr(B_2) &= \Pr(B_2 \mid B_1)\Pr(B_1) + \Pr(B_2 \mid R_1)\Pr(R_1) \\ &= \frac{99}{999} \times 0.1 + \frac{100}{999} \times 0.9 \\ &= 0.1\end{aligned}$$

   But what if we had asked for the probability the nth marble is blue? Using total probability would be pretty complicated.

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.
   We want to find $\Pr(B_2)$.
   By total probability,

$$\begin{aligned}\Pr(B_2) &= \Pr(B_2 \mid B_1)\Pr(B_1) + \Pr(B_2 \mid R_1)\Pr(R_1) \\ &= \frac{99}{999} \times 0.1 + \frac{100}{999} \times 0.9 \\ &= 0.1\end{aligned}$$

   But what if we had asked for the probability the nth marble is blue? Using total probability would be pretty complicated.
   By symmetry, the probability that the n-th marble picked from box 1 is blue with probability 0.1.

Solution (cont.)

3. Let B_1 be the event that first marble is blue, R_1 be the event that the first marble is red, and B_2 be the event that second marble is blue without looking at the color of first marble.
   We want to find $\Pr(B_2)$.
   By total probability,

$$\begin{aligned}\Pr(B_2) &= \Pr(B_2 \mid B_1)\Pr(B_1) + \Pr(B_2 \mid R_1)\Pr(R_1) \\ &= \frac{99}{999} \times 0.1 + \frac{100}{999} \times 0.9 \\ &= 0.1\end{aligned}$$

   But what if we had asked for the probability the nth marble is blue? Using total probability would be pretty complicated.
   By symmetry, the probability that the n-th marble picked from box 1 is blue with probability 0.1.
   All the permutations of the 1000 marbles have the same probability, so the probability that the n-th marble is blue is the same as the probability that the first marble is blue.

Diversify Your Hand

You are dealt 13 cards from a standard 52 card deck. Let X be the number of distinct values in your hand (The 13 possible values are Ace, 2, 3, 4, ..., Jack, Queen, King.) For instance, the hand (A, A, A, 2, 3, 4, 4, 5, 7, 9, 10, J, J) has 9 distinct values.
Calculate $\mathbb{E}[X]$.

Solution

X_i: Indicator that ith value appears in hand.

$X = \sum_{i=1}^{13} X_i \implies \mathbb{E}[X] = \sum_{i=1}^{13} \mathbb{E}[X_i]$ (**linearity**).

Solution

X_i: Indicator that ith value appears in hand.

$X = \sum_{i=1}^{13} X_i \implies \mathbb{E}[X] = \sum_{i=1}^{13} \mathbb{E}[X_i]$ (**linearity**).

$\Pr[X_i = 1] = 1 - \Pr[X_i = 0] =$

$1 - \Pr[\text{card } i \text{ does not appear in hand}]$.

Solution

X_i: Indicator that ith value appears in hand.

$X = \sum_{i=1}^{13} X_i \implies \mathbb{E}[X] = \sum_{i=1}^{13} \mathbb{E}[X_i]$ (**linearity**).

$\Pr[X_i = 1] = 1 - \Pr[X_i = 0] =$

$1 - \Pr[\text{card } i \text{ does not appear in hand}]$. This is $1 - \frac{\binom{48}{13}}{\binom{52}{13}}$.

Solution

X_i: Indicator that ith value appears in hand.

$X = \sum_{i=1}^{13} X_i \implies \mathbb{E}[X] = \sum_{i=1}^{13} \mathbb{E}[X_i]$ (**linearity**).

$\Pr[X_i = 1] = 1 - \Pr[X_i = 0] =$

$1 - \Pr[\text{card } i \text{ does not appear in hand}]$. This is $1 - \frac{\binom{48}{13}}{\binom{52}{13}}$. Then,

$$\mathbb{E}[X] = 13 \Pr[X_1 = 1] = 13\left(1 - \frac{\binom{48}{13}}{\binom{52}{13}}\right).$$