EECS 70 Discrete Mathematics and Probability Theory
Spring 2014 Anant Sahai

Midterm 2

Exam location: 1 Pimentel, back half: SIDs with second-to-last digit 1 or 3

PRINT your student ID: | | | | | | | | | |

PRINT AND SIGN your name: _____, _____ _____
(last) (first) (signature)

PRINT your Unix account login: cs70-_____

PRINT your discussion section and GSI (the one you attend): _____

Name of the person to your left: _____

Name of the person to your right: _____

Name of someone in front of you: _____

Name of someone behind you: _____

Section 0: Pre-exam questions (3points)

1. **What is your favorite song? (1 pt)**

2. **Describe a sense of accomplishment that you have felt and what prompted it. (2pts)**

Do not turn this page until the proctor tells you to do so.

PRINT your name and student ID: _____

Section 1: Straightforward questions (50 points)

You must show work to get credit. You get two drops: do 5 out of the following 7 questions (we will grade all 7 and keep only the 5 best scores). However, there will be essentially no partial credit given in this section. Students who get all 7 questions correct will receive some bonus points.

3. Interpolate (10 points)

Prof. Sahai decides to share his favorite Pokémon among six of the GSIs, but to keep it a secret unless enough GSIs come together. Each potential favorite Pokémon is given a number (Charizard=0, Wartortle=1, Pidgeot=2, Ninetales=3, Arcanine=4, Scyther=5, Jolteon=6) and the favorite is hidden as the constant term (i.e. as usual, the secret is in $P(0)$) in a polynomial, $P(x)$, of degree $d \leq 4$ using $GF(7)$. You manage to acquire the following five points by attending different GSI office hours: $(1,0),(2,6),(3,0),(4,0),(6,0)$. **Use Lagrange Interpolation to find out the secret** (the Prof's favorite Pokémon).

Solution:

$$\begin{aligned}
P(x) &= y_1\Delta_1(x) + y_2\Delta_2(x) + y_3\Delta_3(x) + y_4\Delta_4(x) + y_6\Delta_6(x) \\
&= 0\cdot\Delta_1(x) + 6\cdot\Delta_2(x) + 0\cdot\Delta_3(x) + 0\cdot\Delta_4(x) + 0\cdot\Delta_6(x) \\
&= 6\cdot\Delta_2(x) \\
&= 6\cdot\frac{(x-1)(x-3)(x-4)(x-6)}{(2-1)(2-3)(2-4)(2-6)} \\
&= 6\cdot\frac{(x-1)(x-3)(x-4)(x-6t)}{(1)(-1)(-2)(-4)} \mod 7
\end{aligned}$$

$$\begin{aligned}
P(0) &= 6\cdot\frac{(0-1)(0-3)(0-4)(0-6)}{(1)(-1)(-2)(-4)} \\
&= 6\cdot\frac{(-1)(-3)(-4)(-6)}{(1)(-1)(-2)(-4)} \\
&= 6\cdot\frac{(-3)}{(-2)} \\
&= 9 = \boxed{2 \mod 7}
\end{aligned}$$

The professor's favorite Pokemon is Pidgeot, aka BIRD JESUS.

4. Compute (10 points)

Find $300^{300} \mod 35$.

Solution: I.

The extension of Fermat's Little Theorem says that if p and q are two distinct primes both bigger than 2, then for any non-zero $a \mod pq$ we have

$$a^{(p-1)(q-1)} \equiv 1 \mod pq$$

In our case $p = 5$ and $q = 7$, so we have that $a^{24} = 1 \mod 35$ for any non-zero $a \mod pq$. Therefore

$$\begin{aligned} 300^{300} &= (300^{24})^{12} \cdot 300^{12} \\ &= 1^{12} \cdot 20^{12} \\ &= (400)^6 \\ &= 15^6 \\ &= (225)^3 \\ &= 15^3 \\ &= 225 \cdot 15 \\ &= 15 \cdot 15 \\ &= \boxed{15 \mod 35} \end{aligned}$$

Solution: II.

$$300^{300} = 0^{300} = 0 \mod 5$$

and

$$300^{300} = 20^{300} = (-1)^{300} = 1 \mod 7$$

We are looking for some number $\mod 35$ that is $0 \mod 5$ and $1 \mod 7$. We try 5, 10, 15... ah, 15 works. So $300^{300} = 15 \mod 35$.

PRINT your name and student ID: _____

5. Argue (10 points)

In RSA, if Alice wants to send a confidential message to Bob, she uses Bob's public key to encode it. Then Bob uses his private key to decode the message.

Suppose that Bob chose $N = 77$.

And then Bob chose $e = 3$ so his public key is $(3, 77)$.

And then Bob chose $d = 26$ so his private key is $(26, 77)$.

Will this work for encoding and decoding messages? If not, where did Bob first go wrong in the above sequence of steps and what is the consequence of that error. If it does work, then show that it works.

Solution:
e should be co-prime to $(p-1)(q-1)$.
$e = 3$ is not co-prime to $(7-1)(11-1) = 60$, so this is incorrect, since therefore e does not have an inverse mod 60.

6. Prove (10 points)

Prove the following statement:

If two degree $d \leq n-1$ polynomials $P(x)$ and $Q(x)$ agree at n distinct xs (i.e. For $x_1, x_2, \ldots, x_n$ distinct, $P(x_i) = Q(x_i)$), then they are the same function everywhere else too.

Solution: I.
By property 2 of polynomials, given $d+1$ points, there exists a unique polynomial of degree at most d that goes through those points. In this problem, we are given n points and two polynomials of degree at most $n-1$. So by uniqueness, polynomials $P(x)$ and $Q(x)$ are the same.

Solution: II.
Let

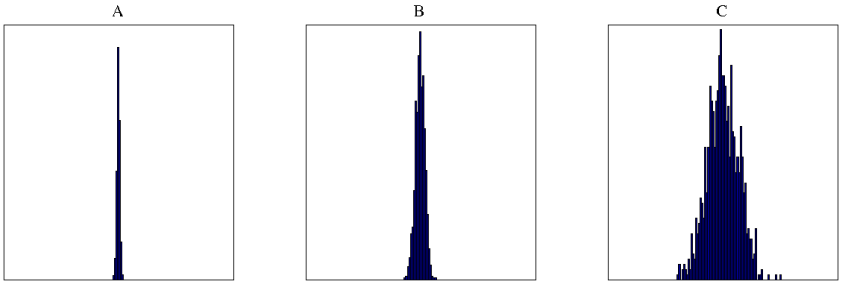
$$R(x) = P(x) - Q(x)$$

Note that because $P(x)$ and $Q(x)$ are of degree at most $n-1$, the degree of $R(x)$ is at most $n-1$. However, $R(x)$ has n roots, because for $x_1, x_2, \ldots, x_n$, $R(x_i) = P(x_i) - Q(x_1) = 0$. By property 1, a non-zero d degree polynomial has at most d roots. Therefore, the only way for $R(x)$ to have n roots is if $R(x) = 0$ for all x. This gives $P(x) - Q(x) = 0$, or $P(x) = Q(x)$.

PRINT your name and student ID: _____

7. Recognize (10 points)

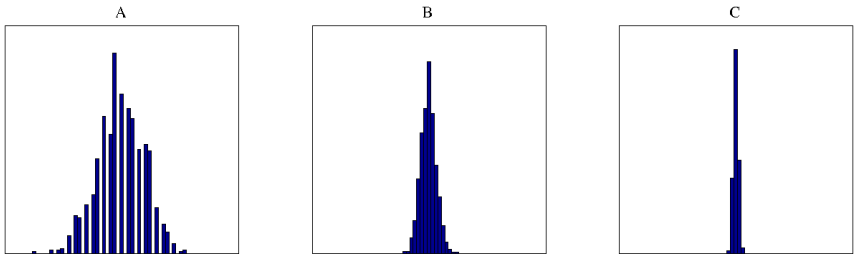
a. I have a large number of biased coins that tend to come up heads 80% of the time, and tails the other 20%. A trial consists of my flipping k such coins, and an experiment consists of 1000 such trials. I do 3 such experiments, with k being 100, 1000, and 10000 respectively. For each experiment, I plot a histogram of the number of heads in each trial, minus $0.8k$. My horizontal axis is the same for all 3 experiments. **Which histogram below corresponds to which value of k?**



Solution: The intuition for this question comes from the virtual lab in Homework 7, problem 1(e) and Discussion 8A problem 1(h). As you probably discovered, the larger the value of k, the wider the distribution becomes. Why is this the case? Let's take smaller numbers to imagine what would happen. Let's say you perform only one trial, what would your distribution look like? You'd expect the histogram to be very narrow – simply because there's really only two options: either 0 heads or 1 head. What if you instead have 10 trials? There are many more possibilities for how many heads you will observe, with some values more likely than others.

Thus, the answer is: **A:** 100, **B:** 1000, and **C:** 10000

b. I do the same experiment as above, but now I plot histograms of the *fraction* of heads (minus 0.8) from each trial. Now **which histogram below corresponds to which value of k?**



Solution: The intuition for this question comes from the virtual lab in Homework 7, problem 1(g). As you probably discovered, the larger the value of k, the narrower the distribution becomes. Let's go back to the case we had before, with 1 flip vs. 10 flips. The chance of flipping 100% heads (or, in other words, flipping heads once), is relatively high at 80% for our biased coin. But what if we had more flips? The chance of getting either of the extremes for the percentage of heads decreases, and you would probably have more trials when doing 10 flips where about 80% of your flips result in heads.

Alternatively, consider the following equation as presented in Note 9, where n denotes the number of flips, ε denotes the error (our deviation from the true proportion of flips we expect to be heads), and δ denotes our confidence level:

$$n = \frac{1}{4\varepsilon^2\delta}$$

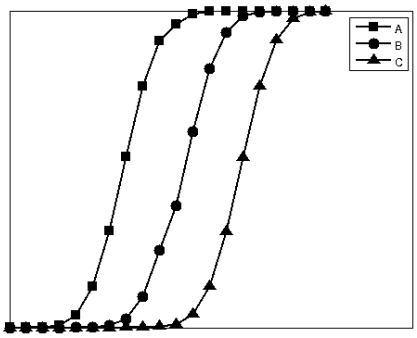
If we fix the confidence level, we have the following relationship:

$$n \propto \frac{1}{\varepsilon^2}$$

From this, as we increase the number of flips, we expect the error to decrease, which corresponds to the narrower histogram observed here.

Thus, the answer is: **A:** 100, **B:** 1000, and **C:** 10000

c. My friend gives me 3 bags full of biased coins: coins in these bags come up heads 40%, 50%, and 60% of the time respectively. Suppose for each q in $0 \leq q \leq 1$, I record the fraction $f(q)$ of trials in which I get at most q fraction of heads when I flip 100 coins drawn from one of these bags. If I plot q (x axis) vs $f(q)$ (y axis), **which curve below corresponds to which bag?**



Solution: This question is effectively asking for you to compare the cliff-face shapes from the virtual labs in the homework. Intuitively, you would expect that the higher the coin's probability of coming up heads p is, the more heads you would expect to see when you flip 100 of them. The curves on the graph are plotting the likelihood that you will see a maximum of q heads as a percentage of trials. This is equivalent to summing up the histogram bars for any value up to q heads in the histograms above. Because we expect more heads with higher p, $f(q)$ would be lower for small values of q.

Thus, the answer is: **A:** 40%, **B:** 50%, and **C:** 60%

PRINT your name and student ID: _____

8. Derive (10 points)

You would like to send a message of length $n > 0$ over a lossy channel that drops (**erases**) packets. If up to a fraction $\frac{1}{4}$ of the *total* number of packets you send get **erased, how many extra packets do you need to send (as a function of n)?**

Solution: We will send $n+k$ packets, of which we want at least n to go through. We know that at least $\frac{3}{4}(n+k)$ packets will make it. Thus,

$$\frac{3}{4}(n+k) \geq n$$

$$n+k \geq \frac{4}{3}n$$

$$\boxed{k \geq \frac{1}{3}n}$$

9. Solve . . . (10 points)

Alice wants to send Bob a message of length 2 in $GF(7)$ over a noisy channel. She knows that at most 1 character will get corrupted when she sends her message. So, she pads her message with 2 extra characters before sending it. (Using standard interpolation-based 0-indexed Reed Solomon codes.)

This is what Bob receives: **A A E G**

What was Alice trying to tell him?

Note: Here, assume that letters correspond to numbers as follows:

$A = 0$
$B = 1$
$C = 2$
$D = 3$
$E = 4$
$F = 5$
$G = 6$

Solution: I.
The four points are (0,0), (1,0), (2,4), (3,6). Let $Q(x) = ax^2 + bx + c$, and $E(x) = x - e$. This gives the four equations:

$$\begin{aligned} c = 0(0-e) &\Rightarrow c = 0 \\ a+b = 0(1-e) &\Rightarrow b = -a \\ 4a-2a = 4(2-e) &\Rightarrow 2a = 8-4e \\ 9a-3a = 6(3-e) &\Rightarrow 6a = 18-6e \end{aligned}$$

You get $a = 2$, $b = -2$ and $e = 1$. This gives $Q(x) = 2x^2 - 2x$ and $E(x) = (x-1)$. The actual polynomial is $P(x) = 2x$, plugging in $x = 1$, we get that the message was $\boxed{\text{AC}}$.

Solution: II.
Bob receives $(0,0,4,6)$. Since Alice's message is length 2, we know that she used a polynomial of degree $2-1 = 1$, i.e. a line. So, we are looking for a line that passes through at least three of the four points $(0,0),(1,0),(2,4),(3,6)$. This line is clearly $y = 2x$ by inspection (draw the points to see it more clearly). So the original codeword was $(0,2,4,6)$ and the original message was "AC".

PRINT your name and student ID: _____

Section 2: True/False (30 points)

For the questions in this section, determine whether the statement is true or false. If true, prove the statement is true. If false, provide a counterexample demonstrating that it is false.

10. Sums (15 points)

Suppose that $n \geq 1$ is a positive integer, $x_1, x_2, \ldots, x_n$ are also nonzero positive integers, and p is a prime. Then

$$(x_1 + x_2 + \cdots + x_n)^p = x_1^p + x_2^p + \cdots + x_n^p \pmod{p}.$$

Mark one: $\boxed{TRUE}$ or FALSE.

Solution: By FLT, if $a \neq 0$, then $a^{p-1} \equiv 1 \mod p$, i.e.

$$a^p \equiv a \mod p$$

And if $a = 0$,

$$a^p \equiv 0^p = 0 = a \mod p$$

So for all a, $a^p \equiv a \mod p$.

Therefore,

$$\begin{aligned}
&(x_1 + x_2 + \ldots + x_n)^p \\
=&(x_1 + x_2 + \ldots + x_n) \\
=&x_1 + x_2 + \ldots + x_n \\
=&x_1^p + x_2^p + \ldots + x_n^p \mod p
\end{aligned}$$

PRINT your name and student ID: _____

11. Distance (15 points)

Let $n \geq 1$ be a positive integer. Let $r \geq 1$ be a positive integer. Consider a polynomial based code in which n character messages are encoded into polynomials of degree less than or equal to $n-1$. The codewords are generated by evaluating these polynomials at $n+r$ distinct points (assume the underlying finite field has more than $n+r$ elements).

Then any two codewords corresponding to different messages must differ in at least $r+2$ places.

Mark one: TRUE or $\boxed{\textit{FALSE}}$.

Solution: Suppose $n = 1$ and $r = 1$. Then the codewords are of length $1+1 = 2$. The statement claims that any two codewords corresponding to different messages must differ in $r+2 = 1+2 = 3$ different places, which is clearly impossible (how can AB and DE differ in 3 places?).

PRINT your name and student ID: _____

Section 3: Free-form Problems (45 points)

12. You knew this was coming... (20 points)

As you know from homework, we can mod polynomials themselves. For this problem, consider formal polynomials (i.e. a degree at most d formal polynomial is something that can be written $\sum_{i=0}^{d} a_i x^i$) with coefficients in $GF(2)$ (i.e. the a_i are 0 or 1 with usual binary math).

(So, for example, x^2+x is the remainder of poly-long-dividing x^4 by x^3+x+1. Meanwhile, the quotient of the same division is just x. This is because $x^4 = x(x^3+x+1)+(x^2+x)$ when all arithmetic on coefficients is performed mod 2.)

Compute the multiplicative inverse of the formal polynomial $x+1$ in mod x^3+x+1. That is, give a formal polynomial $P(x)$ so that $P(x)(x+1) \bmod (x^3+x+1) = 1$.

(e.g. The multiplicative inverse of x is x^2+1 because $x(x^2+1) \bmod (x^3+x+1) = 1$.)

Solution:

Perform the EGCD algorithm for polynomials. Note that all coefficients are in mod 2, so for example, $-2x = 0x$.

x	y	d	a	b	expression
$x+1$	x^3+x+1	1	$-x^2-x$	1	$(-x^2-x)(x+1)+1(x^3+x+1)=1$
$x+1$	$x^3+x+1-(x^3+x^2)-(x^2+x^1)=-2x^2+1=1$	1	0	1	$0\cdot(x+1)+1\cdot 1=1$
0	1	1	0	1	$0\cdot 0+1\cdot 1=1$

Now $(-x^2-x)(x+1)+1(x^3+x+1)=1$, which means

$$\begin{aligned}(-x^2-x)(x+1)+1(x^3+x+1) &\equiv 1 \mod x^3+x+1\\ (-x^2-x)(x+1) &\equiv 1 \mod x^3+x+1\\ (x+1)^{-1} \equiv (-x^2-x) &\equiv x^2+x \mod x^3+x+1\end{aligned}$$

So the multiplicative inverse is x^2+x.

PRINT your name and student ID: _____

13. Magic Command (25 points)

You are a young technomage (one who uses technology to create the impression of magic) who has been asked to help some people on a planet with a fragile new peace treaty. They have a master computer that is connected to a doomsday device capable of blowing up the planet if given the publicly-known command "Magic computer, please blow up the world now." (For the purposes of this problem, feel free to think of this as being a publicly-known magic number like 42).

The problem is that the computer has no security on it. It just accepts plain text commands.

a. (10 pts) You have been asked to add some security to the system to prevent unauthorized use. You can remove the keyboard, add a tamper-proof-decryption module, and force all keyboard input to go through the decryption module before it goes to the computer. But anyone can walk up to the decryption module to study it because it is in the **public** square. **Please show how you would design such a public module that transforms inputs before feeding them into the computer.** This module should effectively make the magic number that blows up the world into something secret.

You can use modulo math operations as you see fit. (You don't have to reprove anything you have seen in lecture or notes.)

Solution:

An important aspect is to understand what it means to be able to study the module. This means that everyone knows exactly how it works; they can see what is inside it. Hiding information inside is not viable so password protection schemes will not work (as you need to store the password inside it). Solutions assuming that tamper-proof meant that you cannot gain (read-only) access to the module did not get full credit.

That said, the scheme basically requires secret information which is not stored in the computer to be given only to authorized individuals.

Method 1:

Construct an RSA public and private keypair, (N,e) and (N,d). Let 42 be the message. The module will take an input x and feed $x^e \mod N$ to the computer. Then, the new secret is $s = 42^d \mod N$. Clearly, if this is fed to the module, $(42^d)^e \equiv 42^{de} \equiv 42 \mod N$ will be fed to the computer and the world will explode as desired. Any other input will not have the same effect, and computing s without the private key d is difficult.

Method 2:

Another solution is to simply require the user to input d. The computer stores 42^e in it (which everyone knows anyway). Then, the computer takes an input t and performs $(42^e)^t \equiv (42^{et}) \mod N$, and the rest proceeds as before.

Method 3:

Another is an encrypted version of a password protection scheme. Every authorized individual is given a password p (could be the same or different for everyone), but the *encrypted* version of the password (i.e. $p^e \mod N$) is stored in the module. An authorized user enters p to gain access to the system. What the module does is: it checks whether the stored value of p^e matches the encryption of the entered password. If it does, then the individual has access.

b. (15 pts) The people on this planet are divided into two factions: the blues and the golds. Within each color group, there are 4 families. They have agreed on the following behavior:

If at least 2 blue families and at least 2 gold families come together, they should be able to give a valid command to the master computer. In addition, if all the blues agree or all the golds agree, then they should also be able to give a valid command to the master computer. But no other grouping should be able to do it. (e.g. 3 blues and 1 gold *should not* be able to give a command.)

Design a scheme and argue why it works as intended. You can use modulo math operations as you see fit, as well as give pieces of information **secretly** to families. (You don't have to reprove anything you have seen in lecture or notes.)

Solution: Generally, the most familiar way to do this was a heirarchical secret sharing scheme, with two parts. One scheme to ensure that 4 families from the same color together could unlock the secret, and one to ensure that 2 families from one color and 2 from another could do it, with both secure against other combinations.

For 4 families of same color: Generate two degree-3 polynomials $P(x)$ and $Q(x)$ over $GF(p)$ for some large prime p, such that $P(0) = Q(0) = s$, where s is the secret we want to hide. Give each blue family a point on $P(x)$, say $P(1), P(2), P(3), P(4)$, and give each gold family a point on $Q(x)$, say $Q(1), Q(2), Q(3), Q(4)$, making sure not to give out the secret.

For 2 families of each color: Generate three degree-1 polynomials $B(x)$, $G(x)$ and $R(x)$, where $R(1) = B(0)$ and $R(2) = G(0)$ and the families are told this. Furthermore, set $R(0) = P(0)$. Give each blue family a point on $B(x)$, say $B(1), B(2), B(3), B(4)$, and each gold family a point on $G(x)$, say $G(1), G(2), G(3), G(4)$.

Justification: If 4 families from either faction get together, they can combine their points and interpolate to get $P(x)$, since from class $n+1$ points completely determine a degree n polynomial, so 4 points will suffice for the degree-3 $P(x)$. However, any fewer than 4 families will not be able to interpolate $P(x)$, as fewer than 4 points are insufficient to determine a degree-3 polynomial, and will in fact gain no information about the secret, again a fact from discussion.

If 2 families from each faction get together, the blue families can combine their points on $B(x)$ to figure out $B(0)$, and the gold families can figure out $G(0)$. From this, they can figure out $R(0)$, since these two points on $R(x)$ give them enough information, hence they can get the secret. However, if there is only one family from either faction (or 0, for that matter), they will not be able to figure out $B(0)$ or $G(0)$, since 1 point is insufficient to determine a degree-1 poly, and hence will not get enough points to figure out $R(0)$.

By the above arguments, you need at least 4 families from 1 faction to get the secret from the first scheme and at least 2 and 2 families to get it from the second scheme, hence no combination that does not lie in one of these cases (i.e. 3 and 1) works.

Common mistakes were not justifying why the scheme prevents 3 blues and 1 gold from accessing the secret, which came down to some acknowledgement that you *need* $n+1$ points to determine a degree n polynomial. Other combining schemes were possible, like addition of random numbers mod p and giving the random numbers out to families.

PRINT your name and student ID: _____

14. (Optional) Multiplication (20 points)

Suppose you have invented a machine for doing *mod multiplications* extremely fast.

Given a prime p, your invention takes two equal size lists of numbers from $\{0,1,\ldots,p-1\}$ as inputs and returns the element-wise product of the input lists, mod p. Your machine is fast and can accomodate any size lists, but it is also prone to mistakes. More specifically, you know that at most $\frac{1}{3}$ of the results are wrong. For example, suppose $p=29$ and we feed the machine the two lists $(1,2,6)$ and $(5,1,2)$ we might get back output $(5,2,11)$ where $11 \neq 6\times 2$ is a mistake. None of your potential clients is interested in a device which returns false results.

Show that you can augment your machine with Reed-Solomon-like encoding and decoding schemes such that no wrong outputs are ever returned and correct answers are obtained to the m pairs of numbers that you actually want to multiply together. (You will need to ask the machine itself to multiply more than m pairs of numbers to accomplish this.) You can assume you have access to interpolation-based (1-indexed) RS-encoders (parameters like n and k can be adjusted as you would like) as well as Berlekamp-Welch decoders. (Again, internal parameters like n and k can be **independently** adjusted as you would like — in particular, you don't have to use the same n and k that you used for the encoders.)

For concreteness, you can assume that the size of the desired input lists is $m=6$ and that $p=127$. (This is a prime number)

(HINT: first explore what happens with $m=1$ and $m=2$ to get an idea for what is going on.)

Solution: We have two messages each of length n. We would like to construct two codewords such that multiplying the codewords gives a new codeword, which is the encoding of the product of the original messages. Suppose we have two messages $a_1,a_2,\ldots,a_n$ and $b_1,b_2,\ldots,b_n$, and the corresponding polynomials P_1 and P_2, which are each of degree at most $n-1$. Now we have the product of the two messages, $a_1b_1,a_2b_2,\ldots,a_nb_n$, and we want there to be a unique polynomial that passes through these points—P_1P_2 works: for all $i=1,2,\ldots,n$, $(P_1P_2)(i)=P_1(i)P_2(i)=a_ib_i$, as desired.

Therefore, if we consider constructing Reed-Solomon codes for the two messages and multiplying them, and then recovering the polynomial and evaluating it, the message that we will get back will be precisely the product of the two messages. So, we only need to ensure that we have enough extra points. However, since P_1 and P_2 are of degree $n-1$, P_1P_2 is a polynomial of degree at most $2(n-1)$. Then, we need $2(n-1)+1=2n-1$ points to recover the polynomial P_1P_2. Therefore, if we let $m=2n-1$, and we send $2k$ additional points, we must have $m+k$ points received without any errors.

If $\frac{1}{3}$ of the results are wrong, we need to send enough extra points to make sure the $\frac{2}{3}$ that survive will be enough to recover: $\frac{2}{3}(m+2k)\geq m+k$, so

$$\begin{aligned}
&\Rightarrow \quad m+2k \geq \frac{3}{2}m+\frac{3}{2}k \\
&\Rightarrow \quad \frac{1}{2}k \geq \frac{1}{2}m \\
&\Rightarrow \quad k \geq m.
\end{aligned}$$

Therefore, we want to send $m+2k=3m=3(2n-1)$ points. To summarize, we encode both messages into polynomials, evaluate a total of $3(2n-1)$ points for each, feed these pairs into the machine, and recover the "message" to find our products.

Note: Students can get full credits if answering $3(2n-1)$ total points, but the minimum number of points to be evaluated is $\max(6n-7,1)$. This can be obtained by the following observation:

- $m=1$: we add 0 additional point, and there is at most $\lfloor \frac{1}{3} \rfloor = 0$ error, so the total number of points is 1.
- $m=2$: we add 0 additional point, and there is at most $\lfloor \frac{2}{3} \rfloor = 0$ error, so the total number of points is 2.
- $m=3$: we add 2 additional points, and there is at most $\lfloor \frac{5}{3} \rfloor = 1$ error. We can correct it because we have 2 additional points, so the total number of points is 5.
- $m=4$: we add 4 additional points, and there are at most $\lfloor \frac{8}{3} \rfloor = 2$ errors. We can correct them because we have 4 additional points, so the total number of points is 8.
- $m=5$: we add 6 additional points, and there are at most $\lfloor \frac{11}{3} \rfloor = 3$ errors. We can correct them because we have 6 additional points, so the total number of points is 11.
- $m>5$: we add $2m-4$ additional points, and there are at most $\lfloor \frac{3m-4}{3} \rfloor = m-2$ errors. We can correct them because we have $2m-4$ additional points, so the total number of points is $3m-4$. If we only add $2m-5$ additional points, then there are at most $\lfloor \frac{3m-5}{3} \rfloor = m-2$ errors. We cannot correct them because we have only $2m-5$ additional points (we need 2(m-2) points).

From the observation, we can get: given m, we need $\max(3m-4,1)$ total points. Plugging in $m=2n-1$, we get $\max(6n-7,1)$ total points.