μ - Games

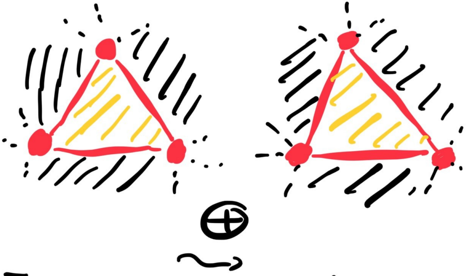
Mathematics Utrecht

Utrecht University

16 December 2022

Problem 1: Bread Mania

- Start by finding a triangulation (or planar surface graph) of a sphere and torus.
- Inductively construct other surfaces by pasting torus onto a graph.
- Notice a pattern: $\chi(S) = 2 - 2g$, where g is the number of holes.



$$\left.\begin{aligned} F_\oplus &= F_1 + F_2 - 2 \\ E_\oplus &= E_1 + E_2 - 3 \\ V_\oplus &= V_1 + V_2 - 3 \end{aligned}\right\} \quad \chi(X_1 \oplus X_2) = \chi(X_1) + \chi(X_2) - 2$$

Problem 2: Weierstrass Fixed Point

$$f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x)$$

- Uniform convergence, so f is continuous.
- When computing f numerically, ensure left over sum $\sum a^n < \epsilon$.

Problem 2: Weierstrass Fixed Point

- Reformulate the problem: $g(x) = f(x) - x = 0$.
- $g(0) = f(0) > 0$ and $g(\pi) = f(\pi) - \pi < 0$.
- Start with $a = 0$ and $b = \pi$.
- IVT states g has a zero somewhere between a and b.

Problem 2: Weierstrass Fixed Point

- Reformulate the problem: $g(x) = f(x) - x = 0$.
- $g(0) = f(0) > 0$ and $g(\pi) = f(\pi) - \pi < 0$.
- Start with $a = 0$ and $b = \pi$.
- IVT states g has a zero somewhere between a and b.

Problem 2: Weierstrass Fixed Point

- Binary search!
- Inductively: $c = \frac{a+b}{2}$
- If $|g(c)| < \epsilon$, we are done.
- If $g(c) > 0$, set $a = c$.
- If $g(c) < 0$, set $b = c$.
- Shape of f ensures convergence.

Problem 2: Weierstrass Fixed Point

- Binary search!
- Inductively: $c = \frac{a+b}{2}$
- If $|g(c)| < \epsilon$, we are done.
- If $g(c) > 0$, set $a = c$.
- If $g(c) < 0$, set $b = c$.
- Shape of f ensures convergence.

Problem 2: Weierstrass Fixed Point

- Binary search!
- Inductively: $c = \frac{a+b}{2}$
- If $|g(c)| < \epsilon$, we are done.
- If $g(c) > 0$, set $a = c$.
- If $g(c) < 0$, set $b = c$.
- Shape of f ensures convergence.

Problem 2: Weierstrass Fixed Point

- Binary search!
- Inductively: $c = \frac{a+b}{2}$
- If $|g(c)| < \epsilon$, we are done.
- If $g(c) > 0$, set $a = c$.
- If $g(c) < 0$, set $b = c$.
- Shape of f ensures convergence.

Problem 2: Weierstrass Fixed Point

- Binary search!
- Inductively: $c = \frac{a+b}{2}$
- If $|g(c)| < \epsilon$, we are done.
- If $g(c) > 0$, set $a = c$.
- If $g(c) < 0$, set $b = c$.
- Shape of f ensures convergence.

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.

Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.

Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.

The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j - th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty}(m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.
Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.
Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.
The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j-th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty}(m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.
Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.
Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.
The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j - th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty}(m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.

Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.

Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.

The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j - th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty}(m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.
Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.
Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.
The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j - th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty} (m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Exercise 3:

The first pirate takes: $m \cdot \frac{p}{q}$.
Then the second pirate takes $(m - m \cdot \frac{p}{q}) \cdot \frac{p}{q}$.
Then what is left is $m - m \cdot \frac{p}{q} - (m - m \cdot \frac{p}{q}) \cdot \frac{p}{q} = m(1 - \frac{p}{q})^2$.
The i-th time a pirate gets money, he/she will get $(m - (1 - \frac{p}{q})^{i-1})\frac{p}{q}$. The $j - th$ pirate (call the first pirate pirate 0) will get:

$$\sum_{i=0}^{\infty}(m(1 - \frac{p}{q})^{n \cdot i + j})\frac{p}{q}$$

This is a geometric series, so it equals

$$\frac{\frac{mp}{q} \cdot (1 - \frac{p}{q})^j}{1 - (1 - \frac{p}{q})^n}.$$

We can rewrite:

$$\frac{mp(q-p)^{n-(j+1)}}{q^n - (q-p)^n}$$

Problem 4: Sturm-Liouville

$$L = \frac{\mathrm{d}}{\mathrm{d}x}\left((1-x)^2 \frac{\mathrm{d}}{\mathrm{d}x}\right)$$

- The eigenfunctions are polynomials of different degrees i.e. p_n.
- Use x^n to find eigenvalue of p_n.

$$Lx^n = -n(n+1)x^n + n(n-1)x^{n-2}$$

- So, we must have $Lp_n = -n(n+1)p_n$.
- Comparing the x^d terms on both sides, we see

$$-d(d+1)a_d + (d+2)(d+1)a_{d+2} = -n(n-1)a_d$$

- Thus, we get $a_d = \frac{(d+2)(d+1)}{d(d+1)-n(n-1)} a_{d+2}$

Exercise 5: Be Squareful

- Find two squareful numbers $< 2 \cdot 10^6$ that differ by k.
- Brute forcing all $a < 2 \cdot 10^6$ does not work; testing for squarefulness is slow.
- Solving a Pell-equation also does not work; note that the given Pell equation does not find all pairs of consecutive squareful numbers.

Exercise 5: Be Squareful

- Find two squareful numbers $< 2 \cdot 10^6$ that differ by k.
- Brute forcing all $a < 2 \cdot 10^6$ does not work; testing for squarefulness is slow.
- Solving a Pell-equation also does not work; note that the given Pell equation does not find all pairs of consecutive squareful numbers.

Exercise 5: Be Squareful

- Find two squareful numbers $< 2 \cdot 10^6$ that differ by k.
- Brute forcing all $a < 2 \cdot 10^6$ does not work; testing for squarefulness is slow.
- Solving a Pell-equation also does not work; note that the given Pell equation does not find all pairs of consecutive squareful numbers.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and p^2n if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as a^2b^3 and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and $p^2 n$ if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as $a^2 b^3$ and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and p^2n if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as a^2b^3 and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and $p^2 n$ if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as $a^2 b^3$ and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and p^2n if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as a^2b^3 and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and p^2n if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as a^2b^3 and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 5: Be Squareful

- Instead: Iteratively construct squareful numbers using a priority queue.
- Pushing to an array does not work as we need the *smallest* solution.
- For each element n, add pn if $p \mid n$ and $p^2 n$ if $p \nmid n$.
- From unique factorization, we get each squareful number exactly once.
- Keep track of squareful numbers and for each n, check if $n - k$ was squareful.
- Many other potential solutions, such as sieving all squareful numbers $< \cdot 10^6$ or writing each squareful number as $a^2 b^3$ and using this decomposition in a priority queue.
- Cheese: Precomputing all squareful numbers locally and storing them in an array.

Exercise 6: Mischievous Moles

- Find the expected time T before some random process returns to its initial state.
- Finding the pattern is hard. Either do a lot of small cases (by programming a simulation), or
- Use theory on Markov chains.

Exercise 6: Mischievous Moles

- Find the expected time T before some random process returns to its initial state.
- Finding the pattern is hard. Either do a lot of small cases (by programming a simulation), or
- Use theory on Markov chains.

Exercise 6: Mischievous Moles

- Find the expected time T before some random process returns to its initial state.
- Finding the pattern is hard. Either do a lot of small cases (by programming a simulation), or
- Use theory on Markov chains.

Exercise 6: Mischievous Moles

- Define the problem as a Markov chain.
- States are all possible carrot permutations.
- Transitions are all possible carrot switches by a mole.
- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.

Exercise 6: Mischievous Moles

- Define the problem as a Markov chain.
- States are all possible carrot permutations.
- Transitions are all possible carrot switches by a mole.
- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.

Exercise 6: Mischievous Moles

- Define the problem as a Markov chain.
- States are all possible carrot permutations.
- Transitions are all possible carrot switches by a mole.
- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.

Exercise 6: Mischievous Moles

- Define the problem as a Markov chain.
- States are all possible carrot permutations.
- Transitions are all possible carrot switches by a mole.
- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.

Exercise 6: Mischievous Moles

- Define the problem as a Markov chain.
- States are all possible carrot permutations.
- Transitions are all possible carrot switches by a mole.
- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.

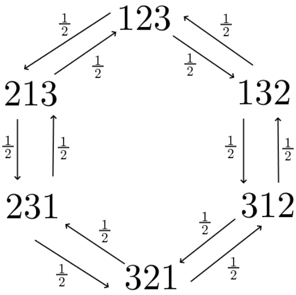


Figure: Example for tunnels $(1, 2)$ and $(2, 3)$.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \dots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \dots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \dots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \dots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \ldots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \dots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \dots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \dots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \dots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.

Exercise 6: Mischievous Moles

- A stationary distribution is $\pi = (\frac{1}{|S|}, \frac{1}{|S|}, \ldots, \frac{1}{|S|})$.
- Theorem: Expected return time to x is given by $\frac{1}{\pi_x} = |S|$.
- Only have to find $|S|$. This is the total number of possible carrot permutations that the moles can create.
- It could be that not all permutations are possible, for instance if we have tunnels $(1, 2)$ and $(3, 4)$.
- All different connected components are independent. If we have connected components of sizes $C_1, \ldots, C_k$, we have $|S| = C_1! \cdots C_k!$.
- Find connected components with your favorite algorithm.