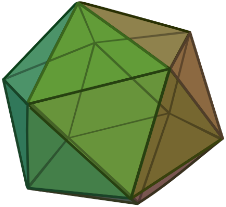
Least Annoying Problem About an Icosahedron

Aug 30, 2024



Where Is the Icosahedron?

Problem

Two strings of 0s and 1s are considered equivalent, if one can be obtained from the other by repeatedly removing or inserting (anywhere in the string) the following substrings: 00, 111, 0101010101. For example, 100110 and 010011 are equivalent, since we can do the following sequence of transformations:

$$100110 \to 1110 \to 0 \to 0111 \to 010011$$

Given 4000 strings of total length at most $5 \cdot 10^4$, find out the groups of equivalent strings among them.

Simpler Problem

Problem

You are given a string s with 10^5 characters, each `a`, `b` or `c`. You must swap exactly two distinct letters to obtain a new string t. How many ways are there to do that in such a way that the string t is good? In this problem we define a good string as follows: empty string is good, if a string u is good then the strings aua, bub, cuc are good as well, and if two strings u and v are good, then their concatenation uv is also good.

Simpler Problem

Solution

Simpler Problem

Solution

- Need to efficiently tell if a string is good

Simpler Problem

Solution

- Need to efficiently tell if a string is good
- Idea: replace each of $\mathtt{a}$, $\mathtt{b}$ and $\mathtt{c}$ with a 2×2 matrix, and associate to a string the product of the corresponding matrices

Simpler Problem

Solution

- Need to efficiently tell if a string is good
- Idea: replace each of `a`, `b` and `c` with a 2×2 matrix, and associate to a string the product of the corresponding matrices
- How do we pick the matrices A, B, C? Need $A^2 = 1$, $B^2 = 1$, $C^2 = 1$, and ideally no other relations.

Simpler Problem

Solution

- Need to efficiently tell if a string is good
- Idea: replace each of `a`, `b` and `c` with a 2×2 matrix, and associate to a string the product of the corresponding matrices
- How do we pick the matrices A, B, C? Need $A^2 = 1$, $B^2 = 1$, $C^2 = 1$, and ideally no other relations.
- Take A, B, C to be the matrix of reflections across three random lines

Simpler Problem

Solution

- Need to efficiently tell if a string is good
- Idea: replace each of `a`, `b` and `c` with a 2×2 matrix, and associate to a string the product of the corresponding matrices
- How do we pick the matrices A, B, C? Need $A^2 = 1$, $B^2 = 1$, $C^2 = 1$, and ideally no other relations.
- Take A, B, C to be the matrix of reflections across three random lines
- Concretely, for a pair of numbers u, v, take

$$A = \frac{1}{u^2 + v^2} \begin{pmatrix} u^2 - v^2 & 2uv \\ 2uv & v^2 - u^2 \end{pmatrix}$$

and the same for B, C with different seeds.

Simpler Problem

Solution

- How do we solve the problem given the way to efficiently tell if any substring is good?

Simpler Problem

Solution

- How do we solve the problem given the way to efficiently tell if any substring is good?
- Divide & Conquer – need to count the number of flips making a string good where one element comes from the left half and the other comes from the right half

Simpler Problem

Solution

- How do we solve the problem given the way to efficiently tell if any substring is good?
- Divide & Conquer – need to count the number of flips making a string good where one element comes from the left half and the other comes from the right half
- For simplicity focus on flipping `a` with `b`. For every `a` in the left side compute the compressed representation of the left side with that `a` changed to `b`, and do the same on the right side. Then for every candidate on the left side count the number of times its inverse occurs on the right side (use matrices modulo prime p instead of floats).

What Did We Just Do?

Group Theory Aside: Definition

- Let's describe this nice solution with scary words

What Did We Just Do?

Group Theory Aside: Definition

- Let's describe this nice solution with scary words
- A *group* $(G, \cdot, e)$ is a set with a binary operation $\cdot$ (usually called multiplication or addition depending on the context) and an identity e, such that
  1. Binary operation is associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
  2. Identity is reasonable: $a \cdot e = e \cdot a = a$
  3. Inverses exist: for any a exists b such that $a \cdot b = b \cdot a = e$.

What Did We Just Do?

Group Theory Aside: Examples

- You already know a bunch of groups

What Did We Just Do?

Group Theory Aside: Examples

- You already know a bunch of groups
- Nonzero (Rational/real/complex/mod p) numbers under multiplication (or addition with 0)

What Did We Just Do?

Group Theory Aside: Examples

- You already know a bunch of groups
- Nonzero (Rational/real/complex/mod p) numbers under multiplication (or addition with 0)
- Invertible matrices with (rational/real/complex/mod p) coefficients under multiplication

What Did We Just Do?

Group Theory Aside: Examples

- You already know a bunch of groups
- Nonzero (Rational/real/complex/mod p) numbers under multiplication (or addition with 0)
- Invertible matrices with (rational/real/complex/mod p) coefficients under multiplication
- Permutations under composition (associative, inverse permutations exist)

What Did We Just Do?

Group Theory Aside: Examples

- You already know a bunch of groups
- Nonzero (Rational/real/complex/mod p) numbers under multiplication (or addition with 0)
- Invertible matrices with (rational/real/complex/mod p) coefficients under multiplication
- Permutations under composition (associative, inverse permutations exist)
- Various symmetries in general (in some sense generalizes the previous two bullet points)

What Did We Just Do?

What Does This Have to Do With Our Problem?

- The definition of problem 1 describes a funky group

What Did We Just Do?

What Does This Have to Do With Our Problem?

- The definition of problem 1 describes a funky group
- The underlying set consists of all words in letters $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, where we identify words different by a series of

$$u \mapsto \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}u\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$$

transformations and concatenations as equivalent

What Did We Just Do?

What Does This Have to Do With Our Problem?

- The definition of problem 1 describes a funky group
- The underlying set consists of all words in letters $\mathtt{a}$, $\mathtt{b}$, $\mathtt{c}$, where we identify words different by a series of

$$u \mapsto \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}u\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$$

  transformations and concatenations as equivalent
- The empty words is the identity, concatenation if is the operation, and reversing the words gives the inverse (why?)

What Did We Just Do?

What Does This Have to Do With Our Problem?

- The definition of problem 1 describes a funky group
- The underlying set consists of all words in letters a, b, c, where we identify words different by a series of

$$u \mapsto \{\mathtt{a}, \mathtt{b}, \mathtt{c}\} u \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$$

  transformations and concatenations as equivalent
- The empty words is the identity, concatenation if is the operation, and reversing the words gives the inverse (why?)
- This group is weird and complicated, we've never seen it before and can't even tell if two elements are equal or not

What Did We Just Do?

What Does This Have to Do With Our Problem?

- The definition of problem 1 describes a funky group
- The underlying set consists of all words in letters a, b, c, where we identify words different by a series of

$$u \mapsto \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}u\{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$$

  transformations and concatenations as equivalent
- The empty words is the identity, concatenation if is the operation, and reversing the words gives the inverse (why?)
- This group is weird and complicated, we've never seen it before and can't even tell if two elements are equal or not
- So we identified to each element of this group a matrix, and matrices we know how to work with, can compute and store in memory

What Did We Just Do?

Group Theory Aside: Homomorphisms

- There of course is a fancy word for this

What Did We Just Do?

Group Theory Aside: Homomorphisms

- There of course is a fancy word for this
- A *group homomorphism* is a function f from a group G to a group H such that
  1. $f(e_G) = e_H$
  2. $f(a \cdot_G b) = f(a) \cdot_H f(b)$

What Did We Just Do?

Group Theory Aside: Homomorphisms

- There of course is a fancy word for this
- A *group homomorphism* is a function f from a group G to a group H such that
  1. $f(e_G) = e_H$
  2. $f(a \cdot_G b) = f(a) \cdot_H f(b)$
- We defined a homomorphism from the weird word group W to the group of all invertible 2×2 matrices with mod p coefficients (or real numbers).

What Did We Just Do?

Group Theory Aside: Homomorphisms

- There of course is a fancy word for this
- A *group homomorphism* is a function f from a group G to a group H such that
  1. $f(e_G) = e_H$
  2. $f(a \cdot_G b) = f(a) \cdot_H f(b)$
- We defined a homomorphism from the weird word group W to the group of all invertible 2×2 matrices with mod p coefficients (or real numbers).
- The latter is much to understand, and a good enough proxy for the purposes of telling elements apart and doing computations. An element is not very likely to map to the identity matrix unless it is already the identity in W.

Still No Icosahedron?

Original Problem

- Recall: two strings of 0s and 1s are considered equivalent, if one can be obtained from the other by repeatedly removing or inserting (anywhere in the string) the following substrings: 00, 111, 0101010101. For example, 100110 and 010011 are equivalent, since we can do the following sequence of transformations:

$$100110 \to 1110 \to 0 \to 0111 \to 010011$$

  Given 4000 strings of total length at most $5 \cdot 10^4$, find out the groups of equivalent strings among them.

Still No Icosahedron?

Original Problem

- Recall: two strings of 0s and 1s are considered equivalent, if one can be obtained from the other by repeatedly removing or inserting (anywhere in the string) the following substrings: 00, 111, 0101010101. For example, 100110 and 010011 are equivalent, since we can do the following sequence of transformations:

$$100110 \to 1110 \to 0 \to 0111 \to 010011$$

  Given 4000 strings of total length at most $5 \cdot 10^4$, find out the groups of equivalent strings among them.
- Any ideas?

Icosahedron!!!

Original Problem

- The problem defines a word group again: relations are exactly given in the problem. If we rename the generators from $0, 1$ to a, b, it is $a^2 = e$, $b^3 = e$, $(ab)^5 = e$.

Icosahedron!!!

Original Problem

- The problem defines a word group again: relations are exactly given in the problem. If we rename the generators from $0, 1$ to a, b, it is $a^2 = e$, $b^3 = e$, $(ab)^5 = e$.
- In fact, turns out this group is finite, and is isomorphic to the symmetries of the **icosahedron**.

Icosahedron!!!

Original Problem

- The problem defines a word group again: relations are exactly given in the problem. If we rename the generators from $0, 1$ to a, b, it is $a^2 = e$, $b^3 = e$, $(ab)^5 = e$.
- In fact, turns out this group is finite, and is isomorphic to the symmetries of the **icosahedron**.
- The same group is also isomorphic to the group of even permutations of length 5, usually denoted by A_5.

Icosahedron!!!

Original Problem

- The problem defines a word group again: relations are exactly given in the problem. If we rename the generators from $0, 1$ to a, b, it is $a^2 = e$, $b^3 = e$, $(ab)^5 = e$.
- In fact, turns out this group is finite, and is isomorphic to the symmetries of the **icosahedron**.
- The same group is also isomorphic to the group of even permutations of length 5, usually denoted by A_5.
- But in contest we don't know any of this trivia, and how does it even help? We just want an AC!

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique
- We just need to find some target group where our relations can be satisfied without incurring too many spurios relations

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique
- We just need to find some target group where our relations can be satisfied without incurring too many spurios relations
- Finding matrices with order 5 seems hard, but permutations are a natural candidate

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique
- We just need to find some target group where our relations can be satisfied without incurring too many spurios relations
- Finding matrices with order 5 seems hard, but permutations are a natural candidate
- Solution: look for pairs of permutations p, q such that $p^2 = \mathrm{id}, q^3 = \mathrm{id}, (pq)^5 = \mathrm{id}$. Looking for length 5 permutations is enough.

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique
- We just need to find some target group where our relations can be satisfied without incurring too many spurios relations
- Finding matrices with order 5 seems hard, but permutations are a natural candidate
- Solution: look for pairs of permutations p, q such that $p^2 = \mathrm{id}, q^3 = \mathrm{id}, (pq)^5 = \mathrm{id}$. Looking for length 5 permutations is enough.
- Replace 0 with p, 1 with q, and compute the compositions for all the strings.

Icosahedron!!!

Original Problem

- Well, we can use the same homomorphism technique
- We just need to find some target group where our relations can be satisfied without incurring too many spurios relations
- Finding matrices with order 5 seems hard, but permutations are a natural candidate
- Solution: look for pairs of permutations p, q such that $p^2 = \text{id}, q^3 = \text{id}, (pq)^5 = \text{id}$. Looking for length 5 permutations is enough.
- Replace 0 with p, 1 with q, and compute the compositions for all the strings.
- Do this for a few pairs in case of collisions, since there are not that many permutations.