MA3K7 Rubric (Mini Project)

Rizwan Tariq (2107025)

A Game with the Strings:

In a big bowl, there are a large number of strings all jumbled up. The strings are of the same length.

You cover your eyes and reach into the bowl of strings. You randomly choose two ends and tie them together. You repeat this process of tying two ends a number of times. You then remove your blindfold and see that some closed string loops have been formed.

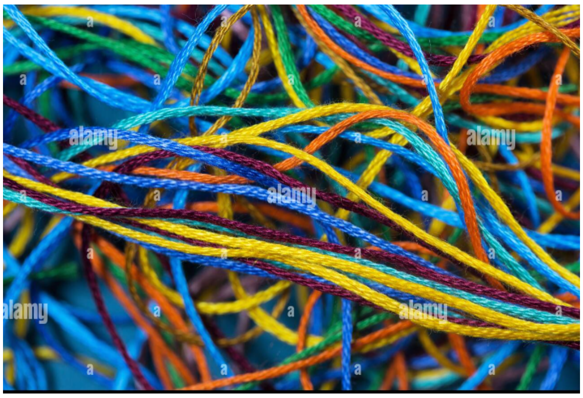


Figure 1: A cool picture of tangled strings

1 Entry

I Know

Introduce

This problem seems a lot more hefty than the previous problems that I have faced. Its almost a little overwhelming as most of the other problems that I have dealt with haven't had much to delve deep into but this problem on the other hand seems to have a lot more components going on compared to the others. This problem briefly describes a game such that we have a bunch of strings in a hat which are all mixed up and we tie 2 random ends together(which could be from the same string since strings have two ends) and we repeat this a number of times and low and behold we have some closed loop as we expected.

I Know

Want

While the closed loops may have been expected due to the nature of the game, what we don't know is the nature of these closed loops. For example:

- How many of these loops will be made?
- Probability of a loop being made?
- Potential size and length of loops made?
- etc

Stuck

There are so many unanswered questions that I wish to dive into in this rubric. When I look at this game, I try to think of a game that is similar to it but nothing comes to my mind.

I would like to specialise in this rubric as the open nature of the problem dictates that we could use a lot of different fields of mathematical studies to investigate this problem. The first thing I think of group theory. Throughout my degree, I have deep dived in the realm of group theory and with this problem, I could try to make this applicable and use it to my advantage which I will explore more in the **ATTACK** phase of this rubric. I can also create a conjecture and try to prove it correct either using python or rigorous proof. I haven't yet thought of what I will exactly conjecture but during the **ATTACK** phase I'm hoping that I will be able to, with more in depth knowledge of the problem, conjecture up something that can be of use or even solve the problem.

WANT

INTRODUCE

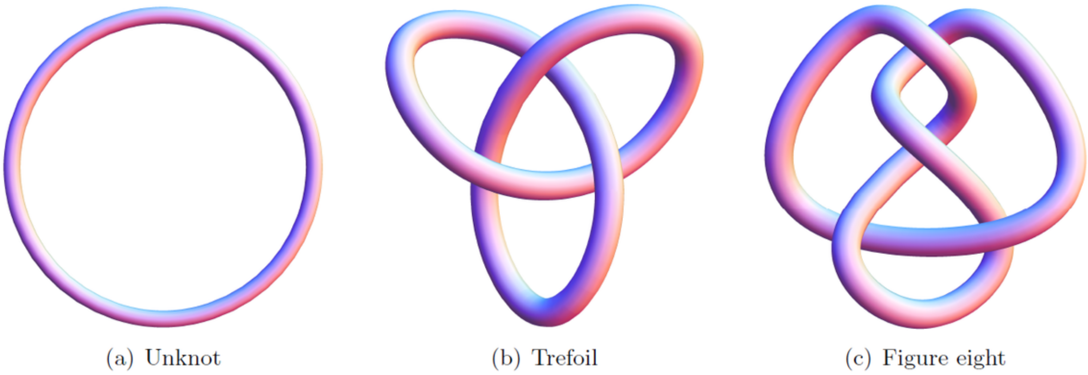


Figure 2: Potential use of Group Theory!!

I also plan to use Python as much as I can to help me speed up this investigation. With a problem as big as this one, I want to be able to not waste time in pointless calculations of visualisation when I can just be using Python to make this investigation more efficient in the long run. A couple of tasks I can do are a simulation of the game in Python, Comparison graphs etc

WANT

A lot of thoughts I just unpacked.. LETS GET STARTED!!!

2 Attack

SPECIALISED SYSTEMATICALLY

Alright, lets start. My initial idea is to play this game initially at very small number of strings in the bag so lets start with just 3 strings in the hat and lets see the possible resulting out comes that come come out of this game.

TRY

Outcome 1

The simplest outcome I can think of is that I grab two ends of the same strings every time and the ending result is I have 3 closed looped strings left in the hat.

Outcome 2

Another simple outcome is that I connect every string to another string at every turn and at the end I connect the two ends of the big string that has been made at the end to create one big loop.

JUSTIFICATION

If we assume that all the strings are all the same colour and same length then there's only one other outcome that could exist.

Outcome 3

We end up with two loops of length ratio 2:1 where one of the loops contains 2 strings connected and the other is just the one string. The sequence of moves is of few forms:

Form 1:

- Tie the two ends of the same string.
- Tie the remaining 2 string together.
- Then tie the the two ends of the big string together to create a big loop.

Form 2:

- Tie 2 string together.
- Then tie the the two ends of the big string together to create a big loop.
- Tie the two ends of the same string.

Form 3:

- Tie 2 string together.
- Tie the two ends of the same single string.
- Then tie the the two ends of the big string together to create a big loop.

STUCK

We can probably do some probability analysis on this game if we made some python code to correspond with the game at hand. My only issue is I don't know how to model the game and it quite hard to find a nice mathematical model that can represent this game.

In the **REVIEW** phase I mentioned that group theory could be of use for us so lets try that!!!

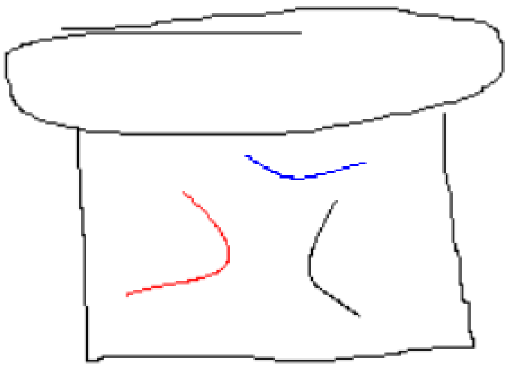


Figure 3: My beautiful drawing of the game on Paint :)

AHA!

Lets take the example above at where there was 3 strings in bag and lets give them labels of 1,2,3:

We will take:

- Red = 1
- Blue = 2
- Black = 3

TRY

I was thinking we could try to model it like a cyclic cycle. For example, we could represent a loop of two strings, lets say red and black, in cyclic notation. So in this case (1 2). My ideology for this is since a closed loop is cyclic if you travel around it, it will repeat itself and just carry on going round and round like a cyclic group.

But...

STUCK

There is no fallible way to even be able to use this to my advantage. For example, i have no clue on how this will help me or aid me at all in this investigation. Which is a bummer because this leaves me clueless on how to proceed.

TRY

Okay lets go back to square one and try to go back to basics. Right now, With the above game of 3 strings, I'm use Python to analyse this game a bit more. In the **REVIEW** phase I asked myself a few questions but I'm going to mainly focus on the one question in this investigation.

Key Question

How many of these loops will be made?

SPECIALISED SYSTEMATICALLY

So lets discuss this!!

STUCK

While I was writing python code I found it very difficult to find a way to represent the game in Python. Its because the game is such visual type game and requires some sort of quantisation to make it fallible to code this in Python. My strategy was:

- Ask for the number of strings that will be put in my hat
- Create a stack with the length of the previous input and use that as my 'hat'.
- Then we had 2 if statements which dictated the difference between picking 2 ends of the same string.
- We then pop off a 'string' if the case of the two ends come from different strings.This is because when you tie two ends of two different strings it turns into one string.
- For the other condition we then iterate a counter when a closed loop is satisfied.
- This is then looped till the array is empty

JUSTIFICATION

BUT WHY

This works because when the loop is formed the string is then useless in the game and so if its popped off from the array then it can get rid of that possibility of being a doubled looped problem.

FURTHER SPECIALISING

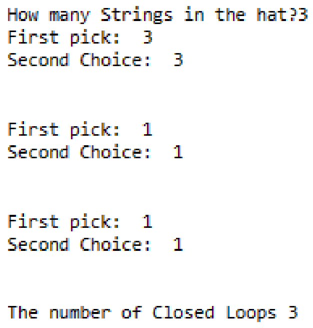
From before, We had 3 possible outcomes before which we discussed and I then tested the code to see if it yielded the same results:

- From **Outcome 1**, we had 3 closed loops.
- From **Outcome 2**, we have one big closed loop.
- From **Outcome 3**, we have a small loop and a bigger loop.

JUSTIFICATION

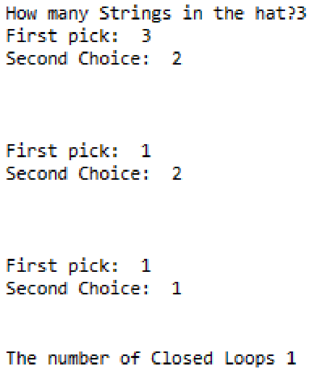
To simplify this we have either 1 loop or 3 loops. By running the program a few times we see a success!!!

With the program I created, I managed to make the number of closed loops for any input n (such that n is the number of strings).



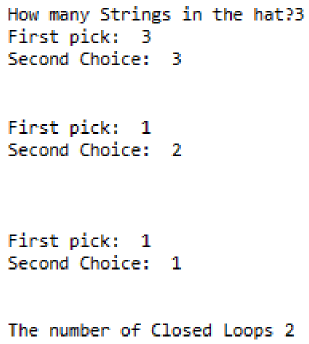
```
How many Strings in the hat?3
First pick:  3
Second Choice:  3


First pick:  1
Second Choice:  1


First pick:  1
Second Choice:  1


The number of Closed Loops 3
```

Figure 4: **Outcome 1**



```
How many Strings in the hat?3
First pick:  3
Second Choice:  2



First pick:  1
Second Choice:  2



First pick:  1
Second Choice:  1


The number of Closed Loops 1
```

Figure 5: **Outcome 2**



```
How many Strings in the hat?3
First pick:  3
Second Choice:  3


First pick:  1
Second Choice:  2



First pick:  1
Second Choice:  1


The number of Closed Loops 2
```

Figure 6: **Outcome 3**

ARTFUL SPECIALISING

So our next step that comes to my mind is to consider how often the certain values of pop up in this game so I have decided to create a histogram/bar chart for around a 1000 games. Below is the bar chart:

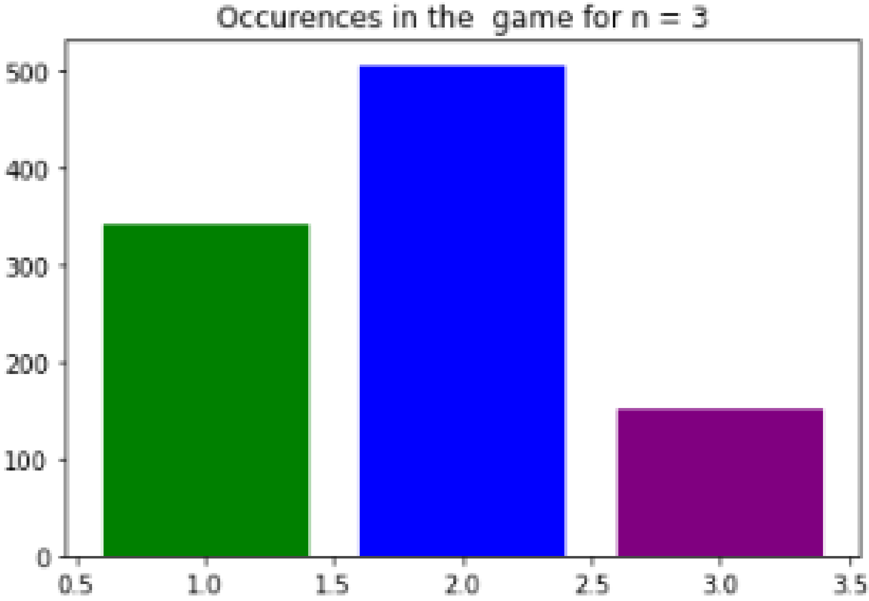


Figure 7: **The bar graph representing the number of closed loop occurrences**

MAYBE

It seems in this case that the number 2 came up a lot more than the other 2 numbers. However, this is just when we consider the numerical approach in this case. This could vary and could be a one off but I'm interested so lets see and run the code again.

TRY

HOWEVER

After multiple running the trend still seems to preserve so this leads to me to,after ages, to conjecture:

CONJECTURE

CONJECTURE

$\mathbb{E}[loops]$ is between 1 and 2 for n = 3

JUSTIFICATION

However, we have been working with n =3 for quite a while while the question didn't really specify how many strings are present in this problem so in order to prove this conjecture I will like to generalised this problem from now on and then substitute the value 3 and see if we can prove this conjecture right!!

Lets do some analysis!

Fortunately, this is actually not too bad. if we let **n** be the number of strings in the bag so by that logic, there are **2n** ends in the bag.

Whenever you grab **1** end of the string in the bag, then there is **2n-1** ends left

JUSTIFICATION

Therefore, the probability of forming a loop on the first move is:

$\frac{1}{2n-1}$

From this, We can say that the expected number of loops is:

$$0 \times \frac{2n-2}{2n-1} + 1 \times \frac{1}{2n-1} = \frac{1}{2n-1}$$

Now, what makes this problem a lot easier to deal with is the fact below:

JUSTIFICATION

No matter what you do, whether it forming a loop or not, you will always have n-1 strings left. This is because if you form a loop in the first move then that loop has no ends left and is useless in the game (I used this fact also when I was crating code for this game). The same can go if you tie two strings together to create a bigger string.

This is very useful for us as that means we don't need to rely on previous moves too much.

JUSTIFICATION

On the second move, We have n-1 strings left in the bag so the expected number of loops is , by using the same thing as before, is $\frac{1}{2(n-1)-1}$.

- For the third move, $\frac{1}{2(n-2)-1}$
- For the fourth move, $\frac{1}{2(n-3)-1}$
- For the fifth move, $\frac{1}{2(n-4)-1}$

and etc.

As we can see, There is a pattern so we can now define:

$$\mathbb{E}[loops] = \sum_{i=1}^{n} \frac{1}{2i-1}$$

And now, if we use n = 3 we yield an answer of:

AHA!

1.533333...
Which matches our conjecture!!

FURTHER SPECIALISING

However, we can carry on this analysis even further so lets carry on!

Looking at the sum above we cant exactly conform it to a single abstract answer as you can with some sums like a geometric or an arithmetic sum.

However, we can use some approximations

We can use a known series like the harmonic numbers to aid us in an estimation. The (2n)th harmonic number is of the following form:

$$H_{2n} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2n}$$

And from my knowledge of other modules we know that this is very close to:

$$ln(2n) + \gamma$$

where γ is the Euler-Mascheroni constant which $\gamma \approx 0.577..$

FURTHER SPECIALISING

The even sums to $\frac{1}{2}(1 + \frac{1}{2} + \cdots + \frac{1}{n})$, which is around $\frac{1}{2}(ln(n) + \gamma)$ and to get the odd terms(which is what we need) we just subtract from the $ln(2x) + \gamma$. If we simplify this, We get:

$$\mathbb{E}[loops] \approx \tfrac{1}{2}(ln(4n) + \gamma)$$

JUSTIFICATION

For reasonably large x this is actually quite accurate and gives very little error. which is good!!

Lets have a loop at the variance of the amount of loops as this pertains quite an interesting answer. As we know, the variance is very closely linked with the expectation but we can actually see the link here below with a few calculations.

JUSTIFICATION

The variance of the number of loops formed on that turn is:

$$\tfrac{1}{2n-1} - (\tfrac{1}{2n-1})^2$$

So

$$Var(loops) = \textstyle\sum_{i=1}^{n}(\tfrac{1}{2i-1} - (\tfrac{1}{2i-1})^2)$$

$$= \mathbb{E}[loops] - \textstyle\sum_{i=1}^{n}(\tfrac{1}{2i-1})^2$$

We know that $\sum_{i=1}^{\text{inf}(\frac{1}{i^2}=\frac{\pi^2}{6}}$ and $\sum_{i=1}^{\text{inf}(\frac{1}{(2i)^2}=\frac{\pi^2}{24}}$

so approximately,

$$Var(loops) = \mathbb{E}[loops] - \tfrac{\pi^2}{8}$$

3 Review

REFLECT

We finally made it to the end of this rubric. This was a very dense problem thin which I got stuck so many times but eventually we pushed through and made it to the end. During the **ENTRY** phase, I had so many thoughts through my head on how to tackle this problem and how to actually make an approach in this problem. There was so many ways I could've approached this problem. I had a lot of questions that I wanted to answer for example the probability, size of loops,etc.

EXTEND

Due to this, in this **ENTRY** phase I wanted to really specialise in a specific area of mathematics so I can really delve deep into that area and use it in this investigation. With the thoughts of loops, knots etc, I thought of the potential use of group theory to maybe aid me in this investigation. I also wanted to really use Python to my advantage and make use of it to my best advantage for the use of graphs etc.

Then we move onto the **ATTACK** phase where we start to actually 'attack' the problem and start the investigation, I initially try to visualise the problem in my head and think of all the outcomes of a possible game for a fixed number of strings in the game. In this case (for the sake of simplicity) we set the number of strings for n =3 as n=2 I thought was too simple and n= 1 is trivial case so we then progress. We then dissected the problem to find every possible outcome and every possible form of the outcome.

CHECK

I then tried to use the group theory approach that I proposed in the **ENTRY** phase and tried to quantize the problem. But this was unfortunately very unsuccessful which was quite a bummer as that was one of the main approach I was really wanting to work. This led me getting really stuck on this problem and didn't really know what to do from this point onwards so i did some deep thinking and i decided to just start with the basics again and use the skills from precious rubrics to try to come up with something that will give me an avenue to find something.

EXTEND

I decided to specialise in one of the questions I asked in the **ENTRY** phase which was the number of loops in the game. I then made a python code of the game and then did some simulations with the number 3 and see if I could get the number of loops as a viable output and if it matched with the outcomes I made earlier in the **ATTACK** phase. Which I found was successful. I then created a bar chat of the results distribution of the problem which then allowed to conjecture something.

From this, I conjectured something and then set out to prove and justify the conjecture which was also success. This also allowed me to further specialise and go even deeper into the problem and find stuff like Variance etc.

REFLECT

Overall, After a very tiring investigation, I'm finally finished. What went well was the constant progressing despite me constantly getting stuck all the time. The fact that I still pushed forward was a plus. What I could've done better was maybe thinking out my approaches more beforehand to avoid me getting stuck all the time. I felt that I learnt a lot more about how to deal with being stuck in this rubric so i definitely feel that this will be very helpful in the future when I need to use my problem solving skills.

Overall, this was very fun!!

Supplementary material

Below is attached my GitHub repository where my code resides:
`https://github.com/rizwan3254/PS-with-yth`