Techsoc Freshie Roadmap

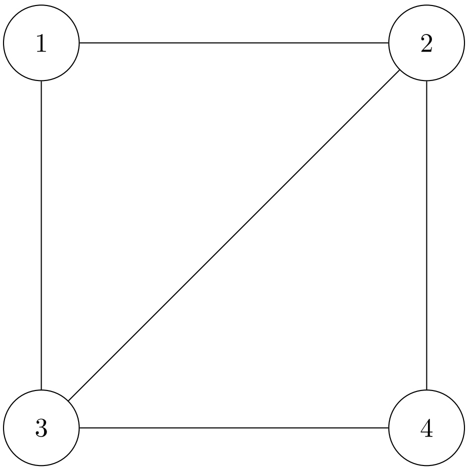
Mathematics Club x Programming Club

Instructions

- Welcome to the first session of the Freshie Roadmaps! The goal is for you to have as much fun as possible.
- All 8 problems are worth 10 marks and the bonus problem is worth 5 marks.
- Try all questions even if you can't solve them completely. Write your answers clearly and concisely.
- Don't use unfair means like the internet because all of you are here to learn and enjoy.

Problem 1: Unsteady State

An ant walks along a square marked as below. At any vertex the ant has equal probability to move in any direction. The ant starts at 1 and walks randomly.After a long time, what is the chance that you find the ant at 2?



Problem 2: To be, or not to be

A machine can either be working or broken on a given day.If the machine is broken, it will be fixed and start working on the following day. The probability that the machine works on a particular day, given that it was working the previous day, is $\frac{1}{3}$. However, if the machine has been working for 3 days in a row, it will break down on the next day. Find the steady-state probability that the machine is working on a given day.

Problem 3: When the chain is too long

You are given a fair 6 sided die and are allowed to roll it for a maximum of 3 times. The payoff is the value that you get on the last roll in Rupees. The ticket to enter the game costs Rs.3 and every subsequent roll costs Rs.1. How many times are you willing to roll the die?

Problem 4: Wait! Is this Markov?

Consider a set of 3 cities in India. The number of roads from one city to another is given in the table below. All the roads are one-way and of unit length. In this matrix, $a_{i,j}$ is the number of roads from city i to city j. Also, the diagonal entries represent the roads that begin and end in the same city. Such a matrix is known as a **adjacency matrix** which stores the number of ways of transition from one state to another.

A **closed loop** is defined as a path that starts and ends at the same city. Let $f(n)$ denote the total number of different closed loops of length n from all the cities.

$$A = \begin{bmatrix} 3 & 15 & 6 \\ 1 & 5 & 2 \\ 2 & 10 & 4 \end{bmatrix}$$

Now, consider the following Python program:

```python
sum = 0
count = 1

for i in range(1,n):
    count *= i
    sum += f(i) / count

return sum
```

What is the output of this program as $n \to \infty$?

Problem 5: Deer Freshies

Akshat, a ranger, needs to move the deers of IITM to a safer area due to an incoming horde of freshies. Akshat being weak can only carry a maximum of k weight at a time and for moving the deers faster he can carry exactly 2 at a time as similar to every other human he has 2 hands and being egoistic refuses to carry only one deer.

He needs to save as many deers as he can as soon as possible or get trampled by the horde of freshies. You as a budding programmer need to tell an algorithm so that he can save as many deers as possible while under the constraints. n numbers are given to you, the weights of the deers.

Problem 6: Dumb Programmers

Archita and Sujal are the only ones to be part of the DP1001(Dumb Programming) course in insti. They had a fight over who is more dumber in the course, so they ended up measuring their skill issue by asking questions to each other.

a) Sujal gives Archita two positive integers **a** and **b** and asks her to find the GCD(Greatest Common Divisor) of **a** and **b**.

Archita creates an algorithm with few error(s), can you help her finding the error(s) in the algorithm.

```
– Step 1: Start the algorithm.
– Step 2: Input two positive integers $a$ and $b$.
– Step 3: Set $gcd$ to 1.
– Step 4: Set $x$ to 0.
– Step 5: If $x \geq \min(a, b)$, go to Step 8. Otherwise, go to Step 6.
– Step 6: If $a$ is divisible by x or $b$ is divisible by x, then set $gcd$ to $x$.
– Step 7: Increase $x$ by 1. Go to Step 5.
– Step 8: Output $gcd$ as the greatest common divisor of $a$ and $b$.
– Step 9: End the algorithm.
```

b) Now Archita gives Sujal a parenthesis combination and asks if the combination is valid parenthesis[†]. Archita being hungry after so much thinking, she brings in a Pringles can. Sujal sees the can and gets an idea.

The characteristics of a Pringles can are:

- You can only see the top element in the can.
- You can only remove the top element in the can.
- You can put element only on the top of the given elements already in the can.
- You can check if the can is empty or not.

He takes the Pringles can from her, eats away the remaining chips and makes an algorithm to check valid parenthesis but ends up making error(s), can you help him find the error(s) in the algorithm. It is guaranteed the combination would just have parenthesis and no other characters.

[†] A combination is said to have valid parentheses if every opening bracket from the set { ([(left brackets) has a corresponding matching closing bracket })] (right brackets) of the same type, and the pairs are properly nested. For example ”[)” is not a valid parenthesis but ”[]{()}” is.

```
– Step 1: Start the algorithm.
– Step 2: Check if Pringles can is empty.
– Step 3: Set $i = 1$, set validity = True.
– Step 4: If i $\leq$ no. of characters in the combination, go to Step 5, otherwise go to Step 7.
– Step 5:
    * If the $i^{th}$ character is a left bracket. Put it in Pringles can.
    * Otherwise
        · If the Pringles can is empty, validity = False, go to Step 6.
        · Otherwise, if the top element on the Pringles can is any one of the left bracket, increment
          i by 1, go to Step 4, otherwise validity = False, go to Step 6.
– Step 6: If validity = True, the sequence is a valid Parentheses sequence otherwise it is not a valid
  sequence.
– Step 7: End the algorithm.
```

Problem 7: Fight for Brownie Points

Veer and Sathvik have n brownies they'd like to split between them, so they decided to play a game. All brownies have a cost. The players move in turns starting from Veer.

In each turn, the player chooses one of the remaining brownies and takes it. The game continues until no items are left.

Let us say that A is the total cost of brownies taken by Veer and B is the total cost of Sathvik's brownies. The resulting *score* of the game then will be equal to $A - B$.

Veer wants to maximize the score, while Sathvik wants to minimize it. Both Veer and Sathvik will play optimally.

But the game will take place tomorrow, so today Sathvik, being unsure of his victory, chooses to modify the costs a little. He can increase the costs a_i of several (possibly none or all) brownies by an integer value . However, the total increase must be less than or equal to k. Otherwise, Veer may suspect something. Note that Sathvik **can't decrease** costs, only increase. Hence, Sathvik decides to try for three different values of k.

However, Sathvik is busy plotting his victory and forgot to find What is the minimum possible score he can achieve for each value of k ? Can you do it for him?

No. of items: 11
Cost of items: 19, 86, 15, 47, 32, 25, 29, 6, 31, 53, 32
Value of k: Case 1 : 100, Case 2: 43, Case 3: 56

Problem 8: Sleep >> Attendance

Sujal tried to fix his sleep schedule but only managed it halfway. He now sleeps early but wakes up late, causing him to miss many of his morning classes. He has asked his good friend, Prajwal, to help him out.
Prajwal has devised a plan based on how the insti's attendance system works:

- He will go to CRC to mark attendance. Each trip is instantaneous: he enters at a specific time, registers Sujal's attendance, and leaves immediately.
- A single visit at time T marks attendance for all classes that are ongoing at that exact moment.
- Because each visit is instantaneous, if one class ends at 9:50 AM and the next one starts at 9:51 AM, they cannot be covered by the same trip.

The course codes and timings for Sujal's 10 classes before lunchtime are given below (Time is in 24 hour format):

DP1001	11:00 - 12:00	DA1001	08:46 - 09:42
ME2010	10:10 - 10:26	ME4200	07:56 - 09:25
HS0960	11:45 - 11:55	CS8030	08:24 - 08:51
CH2222	09:00 - 10:00	PH1010	11:35 - 11:45
MA1080	08:56 - 10:50	CY1000	10:32 - 10:45

- What is the minimum number of trips Prajwal has to make to mark attendance for all of Sujal's classes, also mention what could be the optimal time instant for each trip?
- Devise an efficient algorithm to solve this problem.

Bonus Problem! Thinkception

Answer any positive integer between 1 and 100. Let's call your answer x. We will calculate $\bar{x}$, the average of all x values answered by everyone giving this paper right now. If x is between $\frac{\bar{x}}{2} \pm 5\%$, you will get points for this question. Have fun :)

Answer Sheet

Team Name: MCxPC
Participant 1 Name: Mathematics Club
Participant 1 Roll Number: MC23B001
Participant 1 Contact Number: 1
Participant 2 Name: Programming Club
Participant 2 Roll Number: PC23B001
Participant 2 Contact Number: 0

1. 0.3.
   By symmetry, $\pi_1 = \pi_4$ and $\pi_2 = \pi_3$
   $$\pi_2 = \frac{1}{2}\pi_1 + \frac{1}{3}\pi_3 + \frac{1}{2}\pi_4$$
   Sum of the probabilities of steady states =1
   Solving the above $\pi_2 = 0.3$
2. $\frac{13}{22}$.

   Consider states $1, 2, 3, 4$ where 1 represents the day that the machine is broken.
   For first 2 states, $\pi_1 = \pi_2$
   For the next 2 states,
   $$\pi_i = \frac{1}{3}\pi_{i-1}$$
   Solving the above,
   $$1 - \pi_1 = \frac{13}{22}$$
3. 2
   If your rolling only 1 die, the expected payoff is 3.5 If you are rolling 2 dice, the expected payoff is
   $$\frac{1}{6}(4+5+6) + \frac{1}{2}(3.5) = 4.25$$
   If you are rolling 3 dice, the expected payoff is
   $$\frac{1}{6}(5+6) + \frac{2}{3}(4.25) = 4.66$$
   If you roll twice, you pay Rs.4 and should hence stop at that.
4. The number of ways of transition from state i to j in n steps is given by the $(i, j)^{th}$ element in A^n. So, closed loops reside in the diagonal of A^n. Therefore, for any n, the total number of closed loops = trace(A^n)
   $$f(n) = tr(A^n)$$

Now, for any i, count variable stores the value of $i!$. The required sum is

$$\sum_{i=1}^{\infty}\frac{tr(A^i)}{i!} = tr(\sum_{i=1}^{\infty}\frac{A^i}{i!})$$

Now, on squaring this matrix, we see that $A^2 = 12A$. So, $A^n = 12^{n-1}A$

$$tr(\sum_{i=1}^{\infty}\frac{A^i}{i!}) = tr(\frac{A}{12} \times \sum_{i=1}^{\infty} 12^i) = \frac{tr(A)}{12} \times (e^{12} - 1) = (e^{12} - 1)$$

Ans: $e^{12} - 1$

---

5. ans for q 5

6. a) Step 4: x should be initialised to 1 as that is the minimum possible gcd of given numbers also division by 0 would be undefined.
   Step 5: Should be "$x > \min(a, b)$" as $min(x, y)$ can be gcd as well hence loop should go till $min(x, y)$.
   Step 6: Should be $a \bmod x = 0$ and $b \bmod x = 0$ as gcd is a number that is a factor of both the given numbers.

   b) Step 5:

   - If you are putting in a left bracket in the Pringles can, increment i by 1 to move forward in the combination.
   - If you find a right bracket, you have to find the left parenthesis of the same type and not ”any” left bracket.
   - If a matching exists between the topmost left parenthesis and the current parenthesis, you have to remove the top element from the Pringles can and continue.

   Step 6: Before this step, you should check if the Pringles can was empty or not, because if its not empty, there are still unmatched left brackets in the combination which makes it a non valid sequence.

7. The new edge joins vertices 1 and 4.
   The path described in the question is called a Eulerian Cycle (look it up!). To satisfy the given conditions, for every edge (road) used to arrive at a particular vertex (house), there will be a different edge used to leave the vertex. So each vertex must have an even number of edges (so that they can be divided into pairs like this). The only vertices with an odd number of edges are 1 and 4. Joining them results in a graph that has an Eulerian Cycle (one such cycle is $4 - 6 - 5 - 4 - 3 - 6 - 1 - 3 - 2 - 1 - 4$)

8. 5 is the number of trips. The trip instant could be any time in the given ranges in order:

   - 8:46 - 8:51
   - 9:00 - 10:00
   - 10:10 - 10:26
   - 10:32 - 10:45
   - 11:45

The efficient algorithm is as follows:

- Sort the classes by their end time in ascending order.
- Since all classes must be covered, you can cover maximum possible number of classes if you go at the end time of a class. (An interval may also be optimal to cover some classes, but in general it is best to go at the end time)
- So you greedily pick up the end time of the first class, check what other classes can be covered in the current trip - and strike them out.
- The next trip must be at the end time of the next non striked class.
- Continue the same process again until you have strike all classes.

Bonus:
The average was 18.82258065. Points for anyone who answered 9 or 10. If you're interested, here's the data!
In terms of strategy, there's a variation to this question where the allowed range is 0 to 100. Then, it is most advantageous for everyone to pick 0, so that $\bar{x} = 0$ and everyone gets the points. However, what makes this question interesting is that everyone picking the same number always results in everyone losing. So one might expect people to choose a random number, giving an average of 50 (which is why 25 has a peak in the graph). However, if enough people think this way and choose $25 = \frac{50}{2}$, the average shifts to 25 and the new correct answer is 12 or 13. This continues for as long as you can go, but it's likely people will stop at 25. Another factor is the presence of favourite numbers like 13, 37 (Veritasium) and 69, which skew the expected distribution. There's no accurate way (that we know of) to solve this, other than to put your best estimate from what others are answering. However, if you have a strategy, do share it in the group!