CMIMC 2022
TCS Round

INSTRUCTIONS

1. Do not look at the test before the proctor starts the round.
2. This test consists of several problems, which require proofs, to be solved within a time frame of **90 minutes**. There are **300 points** total.
3. Answers should be written and clearly labeled on sheets of blank paper. Each numbered problem should be *on its own sheet*. If you have multiple pages, number them as well (e.g. 1/3, 2/3).
4. Write your team ID on the upper-right corner and the problem and page number of the problem whose solution you are writing on the upper-left corner on each page you submit. Papers missing these will not be graded. Problems with more than one submission will not be graded.
5. Write legibly. Illegible handwriting will not be graded.

Mysterious Cards

You are given a deck of cards, all face down, numbered from 1 to n, where $n = 2^{100} \approx 1.27 \cdot 10^{30}$. You are not allowed to look at the numbers on any of the cards, but, on each step, you may place the cards (face down) in a row from left to right, and you will be told the number of triples of cards whose values are ordered from least to greatest, going from left to right. For instance, if $n = 6$, and we laid down cards with values $4, 1, 5, 2, 3, 6$, from left to right, we would get a result of 6 triples (corresponding to $456, 156, 123, 126, 136, 236$).

Find an algorithm that, in at most k steps, will always allow you to determine the number written on every card. You do not have to be told that all triples in a placement are in order, but you need to have enough information to be certain that you know the number on each card.

Scoring

An algorithm that completes in **at most** k steps will be awarded:

- 1 pt for $k > 10^{100}$
- 10 pts for $k = 10^{100}$
- 30 pts for $k = 10^{61}$
- 50 pts for $k = 10^{33}$
- 80 pts for $k = 2.6 \cdot 10^{30}$
- 100 pts for $k = 1.3 \cdot 10^{30}$

You are allowed to prove multiple bounds, and will receive points for the best bound that is correctly proven. **Please state which bound you are trying to prove above any proof, or you may not be awarded points for that bound.**

Partial points may be awarded for progress towards the next bound, and partial points may be deducted for logical flaws or lack of rigor in proofs. To get full points, you must demonstrate that your algorithm always produces a correct result, and always runs in at most the stated number of moves.

A Dungeon Journey

You have been placed in a dungeon with $n = 2^{100} \approx 1.27 \cdot 10^{30}$ rooms, numbered from 1 to n, each of which is connected to some other rooms by a two-way hallway. You are in room 1, but you don't know what the dungeon looks like. Even worse, going through a hallway (almost) completely wipes your memory! Your task is to gather a complete map of every room and hallway. (A complete map is a list of every pair of rooms connected by a hallway)

Each room has a large blackboard where you can write down anything you want. The next time you visit the room, you'll be able to see what you wrote, despite your memory being wiped. When walking between rooms, you may carry k bits of information (or alternatively, an integer from 0 to $2^k - 1$, inclusive), but all other memory will be forgotten.

When in your current room, you can see:

- The number of hallways leading out of the room (but the hallways are indistinguishable, and you cannot mark them)
- Any information written on the blackboard from previous visits.
- The k bits of information you chose in the previous room.
- The room number (an integer between 1 and n, where room 1 is your starting room)

You must then choose:

- What to write on the blackboard. (there is no limit to how much you can write)
- k bits of information to bring to the next room.

Since the hallways leaving your room are indistinguishable, you pick one to walk through at random! You may assume:

1. Every room is reachable from every other room.
2. Despite choosing randomly, you are lucky enough that you traverse each hallway at least once (in both directions) every n^{n^n} moves.
3. No room has a hallway to itself, or multiple hallways to the same room.
4. Every hallway is two-way; if it goes from A to B then it goes from B to A.

Design an algorithm to eventually create a complete map of the dungeon, **that you know to be correct**, in some room, in a finite number of steps. You will also remember this algorithm when moving between rooms.

Scoring

An algorithm that can map the dungeon carrying at most k bits of information will be awarded:

- 1 pt for $k > 10^{63}$
- 5 pts for $k = 10^{63}$

- 10 pts for $k = 10^{60}$
- 40 pts for $k = 350$
- 50 pts for $k = 250$
- 60 pts for $k = 150$
- $100 - 2(k-1)$ pts for $1 \leq k \leq 20$

You are allowed to prove multiple bounds, and will receive points for the best bound that is correctly proven. **Please state which bound you are trying to prove above any proof, or you may not be awarded points for that bound.**

Partial points may be awarded for progress towards the next bound, and partial points may be deducted for logical flaws or lack of rigor in proofs. To get full points, you must demonstrate that your algorithm always produces a correct result.

Long Knight's Tour

Consider a n by n chessboard with squares labelled $(1,1)$ through (n,n). On this chessboard lies a long knight. The long knight can move from square (x,y) to (x',y') if one of the two following conditions hold:

- $|x-x'|=3$ and $|y-y'|=1$
- $|x-x'|=1$ and $|y-y'|=3$

In essence, it is a normal chess knight, but longer.

A 'tour' of the chessboard is a sequence of squares $S_1, S_2, S_3, \ldots S_n$ such that for all $1 \leq i \leq n-1$ the move from S_i to S_{i+1} is a valid move for a long knight. Such a tour is considered 'complete' if and only if the tour visits each row and column of the chessboard **exactly** once.

			4	
1				
		5		3
	2			

The above picture is an example of a tour on a 5 by 5 grid, but the tour is not 'complete' as Row 4 is visited twice, and Row 3 is visited zero times.

For each positive integer n, determine whether it is possible for a complete tour of an n by n chessboard to exist.

Scoring

- $2k$ pts for resolving the problem for $1 \leq k \leq 15$ values of n
- 35 pts for resolving the problem for infinitely many values of n
- 65 pts for resolving the problem for all but finitely many values of n
- $100-2k$ pts for resolving the problem for all but $1 \leq k \leq 15$ values of n
- 100 pts for resolving the problem for all $n \geq 1$

Please state which bound you are trying to prove above any proof, or you may not be awarded points for that bound.

Partial points may be awarded for progress towards the next bound, and partial points may be deducted for logical flaws or lack of rigor in proofs. To get full points, you must rigorously show proofs for n for which the problem is impossible, and have clear and reasonable ways to show a long knight's tour exists for the n for which it is possible (including but not limited to explicit constructions).