Change log:

- Problem 4 (version 1): noted a discrepancy between the version here and the autojudger version about an additional operation that can be performed on leaves. Statement is unchanged.

This problem set has a total of 6 problems on 3 pages. Written solutions should be submitted to GradeScope before 4:45pm on Wednesday Sep 30.

Your solutions should be submitted according to the guidelines [https://www.cc.gatech.edu/~rpeng/CS4540_F20/ProblemSetGuidelines.pdf](https://www.cc.gatech.edu/~rpeng/CS4540_F20/ProblemSetGuidelines.pdf) In particular:

1. If you choose not to submit a typed write-up, please write neat and legibly.
2. No credit will be given to solutions obtained verbatim from the Internet or other sources, and uploaded codes will be ran through similarity checking software.

1. A domino is a $2 \times 1$ rectangle. A domino tiling of a rectangle is a way to cover all its cells with non overlapping dominos. Give an $O(n)$ time algorithm that computes the number of tillings of an $3 \times n$ rectangle, modulo 10000.

   Auto judge: [https://dmoj.ca/problem/dwite10c3p3](https://dmoj.ca/problem/dwite10c3p3)

2. An independent set of a graph is a (possibly empty) subset of vertices such that no two vertices from the set are adjacent.

   Give an $O(n)$ time algorithm that takes a tree on $n$ vertices, and outputs the number of independent sets mod 1,000,000,007.

   Auto judge: [https://dmoj.ca/problem/dpp](https://dmoj.ca/problem/dpp)

3. Given a tree on $n$ vertices, a path of vertices

   $$u_1, u_2, \ldots, u_k$$

   is parity-alternating if the odd indices on the path have odd vertex numbers, and the even indices have even vertex numbers.

   Compute the number of parity-alternating paths in $O(n)$ time.

   Auto judge: [https://dmoj.ca/problem/dmpg18s4](https://dmoj.ca/problem/dmpg18s4)
4. Given a tree on \( n \) vertices with values at leaves that you want to send to root, and edge capacities initially set to 1.

You have a total budget of \( X \) to increase these capacities: if you spend \( x \) on some edge, its new capacity becomes \( (x + 1)^2 \).

Give an \( O(nX^2) \) time algorithm for finding the maximum amount of goods that can be sent to the root under this budget.

Auto judge: [https://dmoj.ca/problem/ccc10s5](https://dmoj.ca/problem/ccc10s5)

**Note:** We recommend you do this one as written: the auto judge input format allows for extra spaces, so you need to read it using space ignoring parsers (e.g. %s in scanf, or cin with strings).

**Note 2:** The version on the autojudge also allows for an extra operation of increasing the amount of goods on leaves. That does not fundamentally change the structure of the problem, so for the written version you still only need to deal with edge capacity changes.

5. Given a rooted tree on \( n \) vertices, with lengths along the edges, and a certain amount of good at each vertices. We want to mark \( k \) vertices as ‘hubs’ to minimize the total distance that the goods need to travel toward the root until they all get to hubs.

Give an \( O(n^2k^2) \) time algorithm for computing the minimum total cost of such a plan.

Auto judge: [https://dmoj.ca/problem/ioi05p6](https://dmoj.ca/problem/ioi05p6)

**Hint:** when solving a subtree, it’s useful to know how far ‘up’ (toward the root) is the closest hub picked. This can be encoded as part of the DP state.
6. You want to create a closed loop among cell boundaries of a grid graph with \( r \)-by-\( c \) cells. Recall that a grid graph with \( r \)-by-\( c \) cells has a \( r + 1 \)-by-\( c + 1 \) arrangement of vertices. Each vertex is connected by edges to the vertices that are vertically and horizontally adjacent to it. A loop is a simple cycle in this graph.

Some of the grid cells have numbers that specify how many of the edges on their boundary must be used in your loop, while others can be filled arbitrarily.

For example, for the following 5-by-5 cells with constraints

![Grid Graph Example](image)

a valid loop using the boundary edges (which is actually a 6-by-6 grid) is

![Valid Loop Example](image)

Give an \( O(r^2 c^3) \) time algorithm that finds the longest length of a loop that could be created. The length of a loop is the number of vertices in it.

Auto judge: [https://dmoj.ca/problem/pacnw10i](https://dmoj.ca/problem/pacnw10i)

**BONUS:** This problem was initially uploaded with \( r, c \leq 6 \), and has a lot of room for bigger data. If your problem is able to solve all test data within the 1-second time limit on an 20-by-c sized grid, your bonus points will be \( 5 \times (c - 6) \).