Grow

This document is a step-by-step guide to **Grow**. It also mentions the basic game rules. Accompanying this file, you will find a standard template. **The participants are expected to submit their codes in C/C++/Java/Python.**

Grow is a two player strategy-based game, where the aim of a player is to grow his/her own tree while blocking the opponent's. This game is played in a 20*20 (row*column) grid.

Basic Terminologies

1. Seed:

The seed is the starting point of the tree.
Your seed must be placed on the first column (i, 0).
The opponent's seed will be placed on the opposite edge of the grid (i, 19).
Only one branch (trunk) emerges from the seed in the first turn.

2. Branch:

The branch is a line between two neighbouring points.
Example: the line joining (1, 0) and (1, 1).

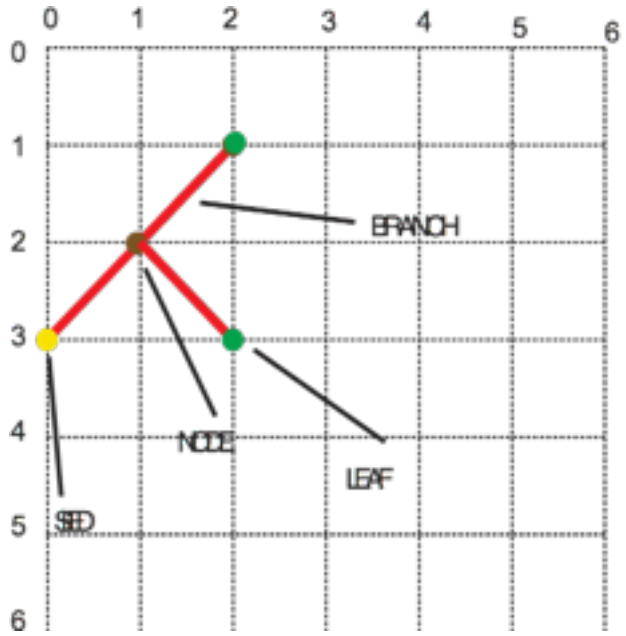
3. Leaf:

It is the open end of a branch which further splits into two distinct branches.
A leaf which **possesses its splitting property** is said to be **active.**
A leaf which **temporarily becomes inactive i.e. cannot be split** is called **a Frozen leaf.**

`

4. Node:

When an active leaf splits into two new branches, the former leaf becomes a node.
A node cannot split. It is just the point from which two branches emerge.



To refer to a leaf use the coordinates (i, j).
For Example: in the above figure, leaf 1 is referenced by the coordinate (1, 2).

Gameplay & Move format

There are three types of moves:

- First Move
- Subsequent Moves
- Termination Move

First move:

Choose a position along the wall for your seed (i_1, 0) and extend your first branch (trunk) from the seed coordinate to (i_2, j_2).

`

Your opponent will do the same on the opposite wall.

Input that your code receives:

20 * 20 array of 9 character strings. Refer to the input format given below.

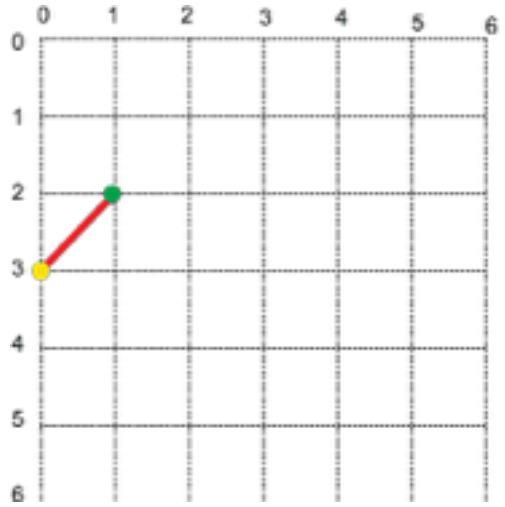
Expected output for the first move:

The player has to enter the seed coordinate and the first leaf coordinate (tip of the first branch i.e. trunk) in the following **string format**:

$$\text{“}i_1,0,i_2,j_2\text{”}$$

where **(i_1, 0)** is the **seed coordinate** (the point where seed is placed) and **(i_2, j_2)** is the **coordinate of the first leaf** (trunk tip).

Example:



(3,0) represents a seed. It grows into a trunk whose tip (2, 1) represents an active leaf. So output string will be “3,0,2,1”.

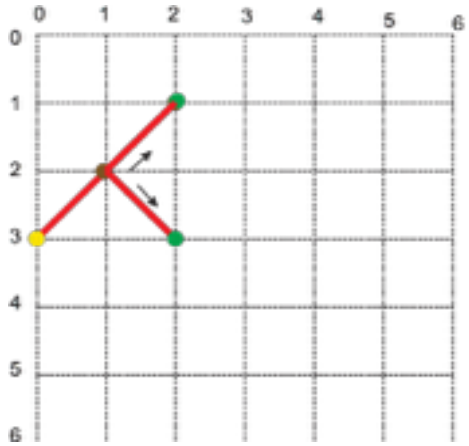
Subsequent moves:

You and your opponent are given alternate chances to plot 2 branches from a chosen active leaf (split the leaf into two new branches).

`

In every move, a chosen active leaf will split into **exactly** 2 new branches, converting the active leaf into a node. The tips of the two new branches represent two new active leaves.

Example :



The active leaf (2, 1) is split into two branches, whose tips (1, 2) and (3,2) represent new active leaves.

Input that your code receives for every move:

String of 3600 characters which is the equivalent of the 20 * 20 array of 9 character strings. It describes the updated grid.

Expected Output for the subsequent moves:

The player has to enter the coordinate of the active leaf which he/she wants to split and the coordinates of the two new leaves in the following **string format**:

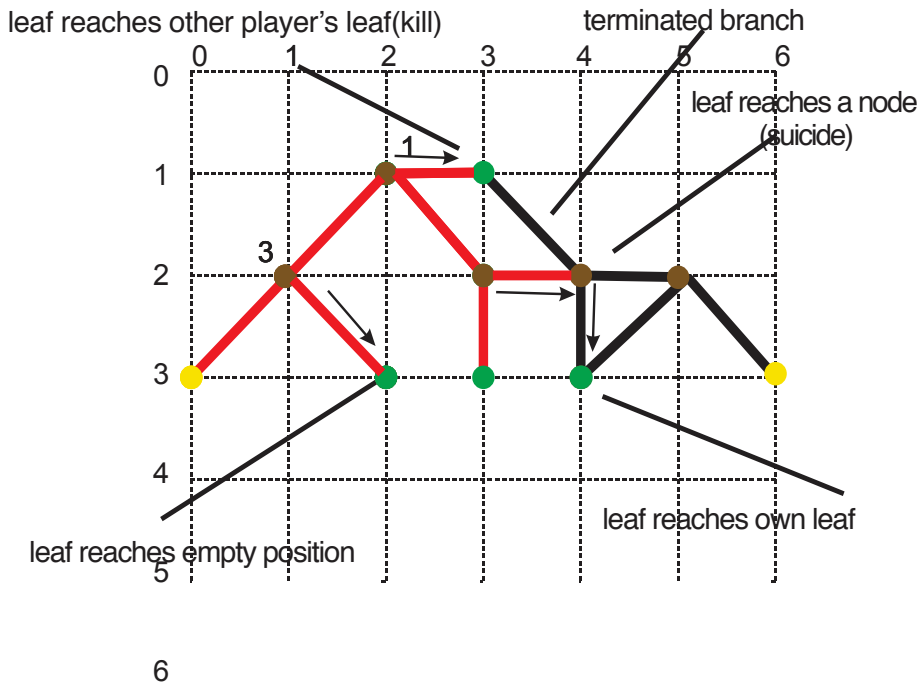
$$\text{“}i_1,j_1,i_2,j_2,i_3,j_3\text{”}$$

where (i_1, j_1) is the **coordinate of the parent leaf** (which you want to split) and **(i_2, j_2), (i_3, j_3)** are the **coordinates of the two new leaves**.

`

VALID MOVES	INVALID MOVES
If new leaf reaches empty position	If new leaf goes beyond the grid
If new leaf reaches your own old leaf (merging)	If player gives same coordinate in the Subsequent Move for the two new leaves((i
If new leaf reaches a node(suicide)	If branch is grown along another branch(retracing)
If new leaf reaches opponent's leaf(kill)	

VALID MOVES



Termination move:

KILL
If new leaf of player 1(terminator) touches leaf of player 2, leaf of player 2 gets terminated (cannot grow further, loses its functionality) and player 1 gets an extra chance (one more move).

In that extra chance, terminator leaf and its sibling cannot be used. They become frozen leaves for that turn only. They can be used in future moves.

SUICIDE
If your new leaf touches a node it will be terminated permanently (cannot grow/will become a node).

MERGING
If your new leaf reaches another one of your own leaves, they henceforth act as one leaf i.e. only two branches can grow from that position.

End Condition

Game will end when:

1. All leaves of a player are terminated.
2. No valid move is left for any player.

Game Scoring

1. For each existing branch, a score of one will be added.
2. However, if a leaf grows into a node, score will not be added for that branch.

`

Player with higher score wins the match!

Judging Criteria

The overall winner of XOdia will be the one with maximum points.

The points are gained as follows:
1. For every match won, the player gets 2 points.
2. The player who loses will get no points (no negative marking).
3. In case of draw, both the players get 1 point each.

Win - 2
Lose - 0
Draw - 1

Input Format

The input to the player is a 20*20 (row*column) 2D array of 9-character strings.

The 2D array will be in the form of a single string format. The 9 character strings of all the coordinates will be passed one after the other, row by row as one complete string i.e. [$R_1R_2R_3$…….R_{20}].
Eg: for a (3*3) 2D array,
“i00000000” “i00000000” “i00000000”
“i00000000” “i00000000” “i00000000”
“i00000000” “i00000000” “i00000000”
The input format is:
“i00000000i00000000i00000000i00000000i00000000i00000000i00000000i00000000i00000000”.

Each co-ordinate’s every property will be stored in it.

`

Initially the entire 20*20 2D array is
A[i][j]="000000000" | 0 < = i < =19, 0 < = j < =19
representing there is nothing present in the grid.
Once the game starts, the 2D array is updated at every turn according to the growth of both the trees.

UPDATED ARRAY

A[i][j]="p00000000" | 0 < = i < =19, 0 < = j < =19

p = { 0,1,2,3,4,5,6 }

if a branch exists from A[i][j] in direction n where n **=** { 1,2,3,4,5,6,7,8 } , A[i][j]**[n]**=1

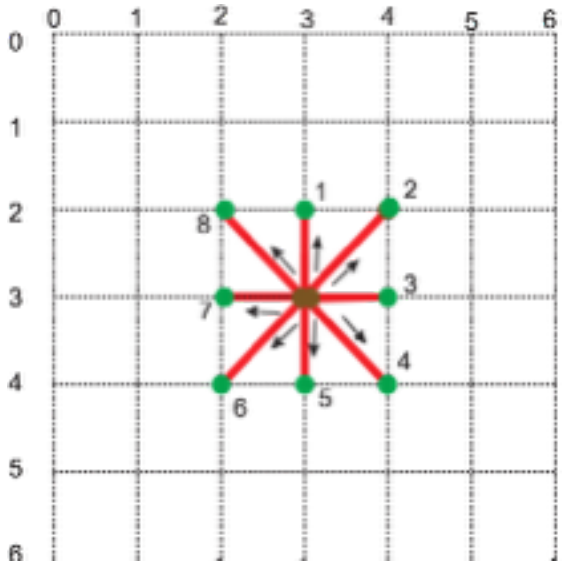
The first character (p) of each 9 character string represents what that coordinate is

0	Nothing present at that point
1	Current player (P1) leaf
2	Opponent (P2) leaf
3	P1 node
4	P2 node
5	P1 frozen leaf
6	P2 frozen leaf

When a leaf splits, it becomes a node. The first character of that point is assigned 3 (you) or 4 (opponent) once that leaf splits representing it is a node.

`

DIRECTION SCHEME

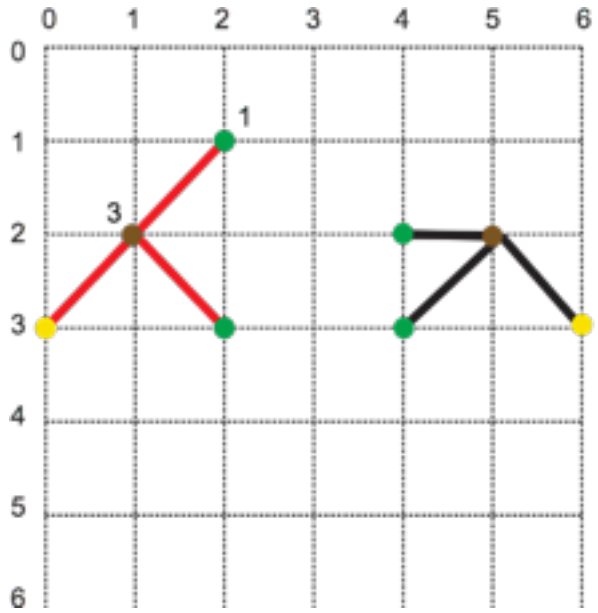


Each direction (vertical, horizontal, diagonal) is represented by a number n from 1 to 8 (n = { 1,2,3,4,5,6,7,8 }). The direction scheme starts from 1 for vertical up direction and increments clockwise as shown in figure.

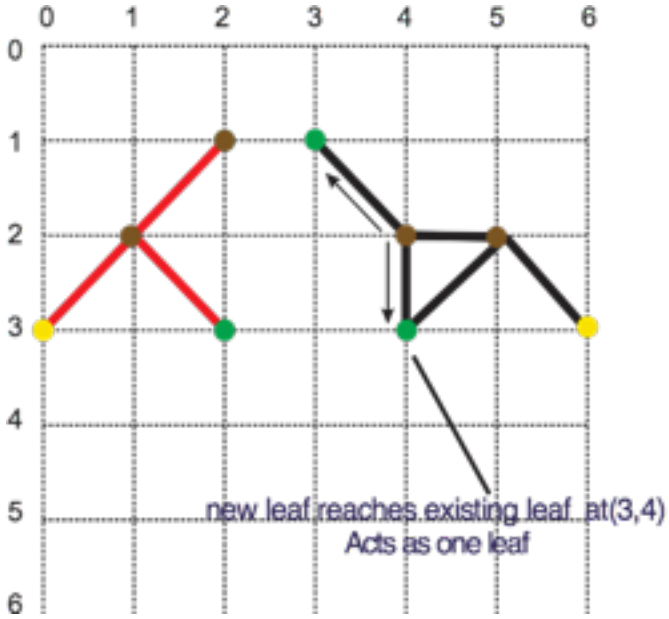
If a branch exists from a point in one or many of the 8 directions, the character representing that direction will become 1.

`

Some Examples

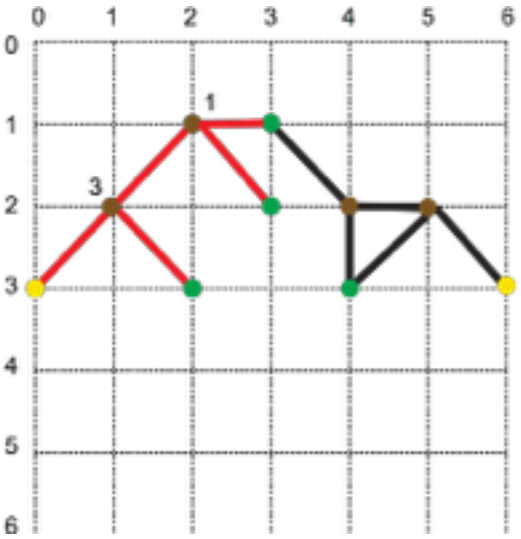


Situation 1

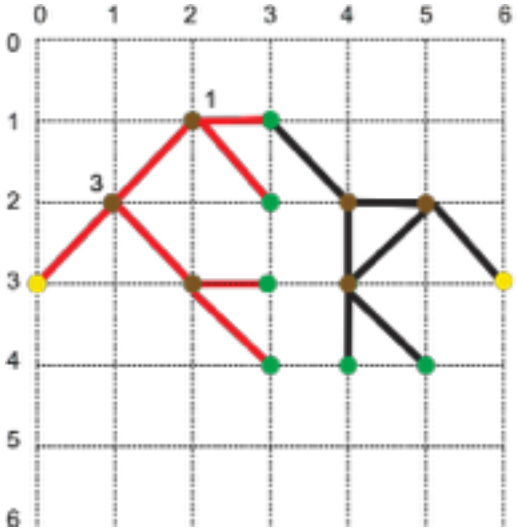


Next move by player 2

Player black has produced two leaves (1,3) & (3,4) from previous leaf(2,4). The new leaf reaches an existing leaf at (3,4), therefore position (3,4) henceforth acts as one leaf.

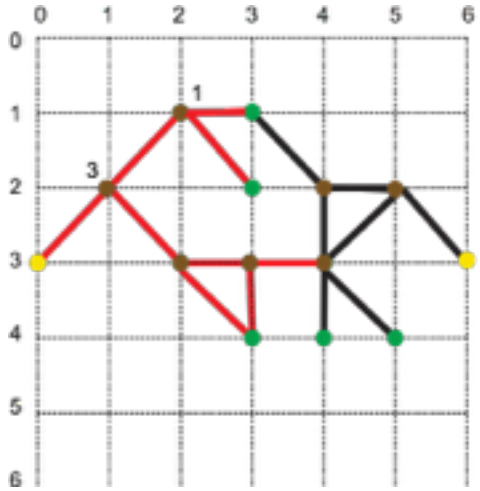


Player red terminates leaf of player black



Player red plays the next move followed by player black

`



Player red terminates his own leaf (suicide) by reaching a node (3,4)

Note

Bot will be disqualified due to any of the following reasons:

- Bot did not respond within 2 seconds of its execution.
- Bot returned an invalid move.
- Syntactical errors in program.
- Excessive resource usage (Bot should consume less then 8MB memory and disk usage must be less then 2MB).
- Any malicious activity encountered in the code (The latest version of the bot would be taken into consideration).

In case of any disputes, the decision of the XOdia team will be final.

MAY THE FORCE GROW WITH YOU!

`