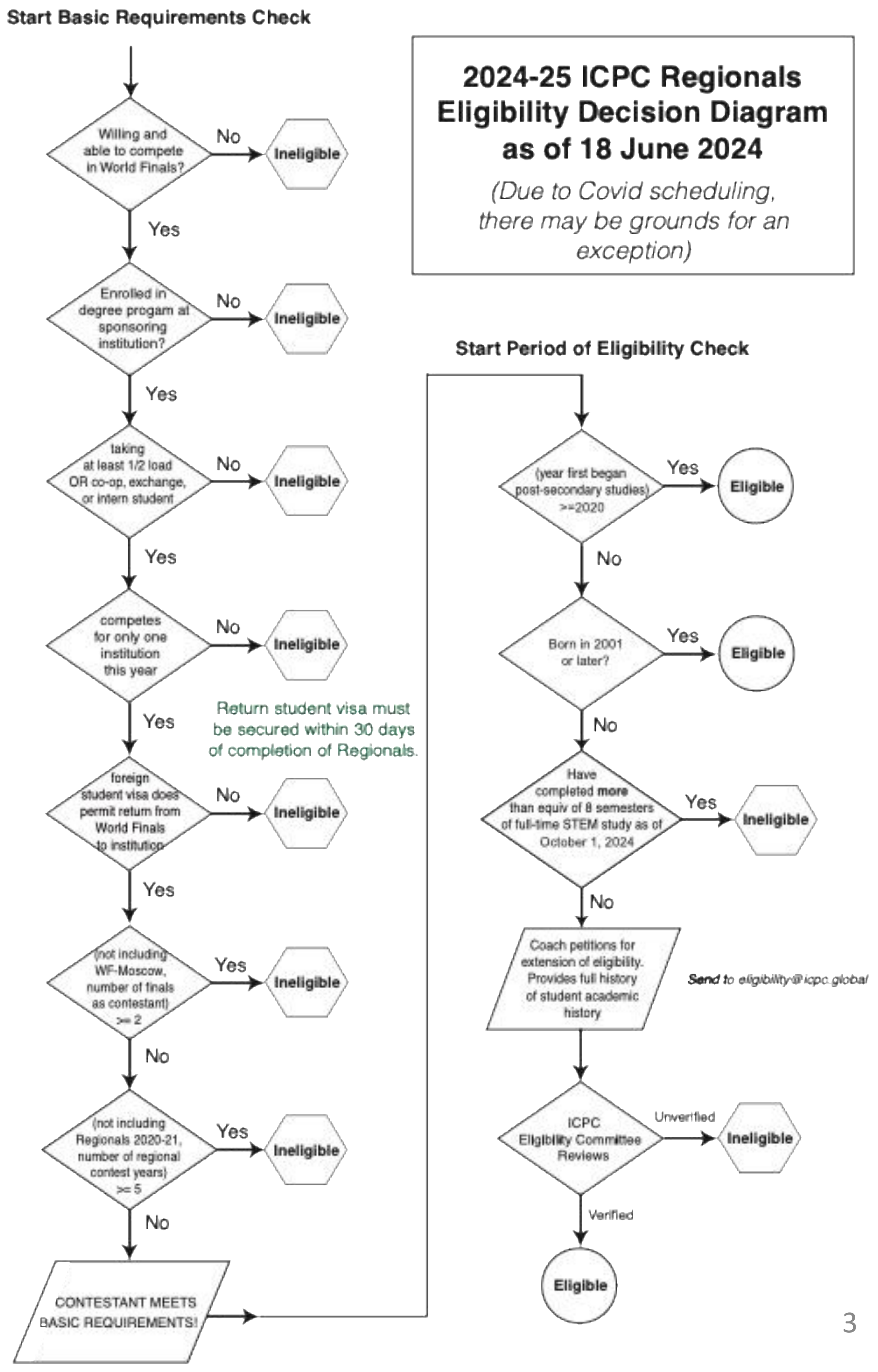
Giri Narasimhan
Programming Team

Fall 2024

Preparing for ICPC Competition … 1

- **North America Qualifier (NAQ)**
  - Oct 5, 2-7 PM on Kattis
- Registered: 3 Teams
  - Asymptotic AC; Binary Brains; Ternary Trios
- Link to contest: https://naq24.kattis.com/
- Info at: https://na.icpc.global/naq
- Registration: https://icpc.global/regionals/finder/North-America-Qualifier

Eligibility


Start Basic Requirements Check
2024-25 ICPC Regionals Eligibility Decision Diagram as of 18 June 2024
(Due to Covid scheduling, there may be grounds for an exception)
Willing and able to compete in World Finals?
No
Ineligible
Yes
Enrolled in degree progam at sponsoring institution?
No
Ineligible
Yes
taking at least 1/2 load OR co-op, exchange, or intern student
No
Ineligible
Yes
competes for only one institution this year
No
Ineligible
Yes
Return student visa must be secured within 30 days of completion of Regionals.
foreign student visa does permit return from World Finals to institution
No
Ineligible
Yes
(not including WF-Moscow, number of finals as contestant) >= 2
Yes
Ineligible
No
(not including Regionals 2020-21, number of regional contest years) >= 5
Yes
Ineligible
No
CONTESTANT MEETS BASIC REQUIREMENTS!
Start Period of Eligibility Check
(year first began post-secondary studies) >=2020
Yes
Eligible
No
Born in 2001 or later?
Yes
Eligible
No
Have completed more than equiv of 8 semesters of full-time STEM study as of October 1, 2024
Yes
Ineligible
No
Coach petitions for extension of eligibility. Provides full history of student academic history
Send to eligibility@icpc.global
ICPC Eligiblity Committee Reviews
Unverified
Ineligible
Verified
Eligible

Teamwork

- 3-person teams, but only one computer
- Teams by Oct 16 to practice as a team
- Team members with complementary skills
- Learn from others
- Learn to debug each others' programs
- No electronic media or devices
- Eligibility:

Problem Repositories & Virtual Judges

- Kattis: https://open.kattis.com/
- VJudge: https://vjudge.net/
- UVa: https://onlinejudge.org/

Topics to Cover

- Dynamic Programming (1D, 2D, …)
- Math Problems (permutations, combinations, probability, counting, Catalan numbers, exhaustive search)
- Graph & Tree Algorithms (DFS, BFS, MST, Dijkstra's, Bellman-Ford, Floyd-Warshall, Topological Sort, Tree Traversal, Tree centroid)
- Advanced Graph Algorithms (network flow, biconnectivity, strong connectivity, …)
- Misc: Geometric Algorithms, String Algorithms, Number Theory Algorithms

Books to help prepare

- Laaksonen A. Guide to competitive programming. Springer International Publishing; 2020.
- Halim S, Halim F, Competitive programming 3. Lulu Independent Publish; 2013.
- Skiena SS, Revilla MA. Programming challenges: The programming contest training manual. Acm SIGACT News. 2003 Sep 1;34(3):68-74.
- Arefin AS. Art of programming contest: C programming tutorials, data structures, algorithms

ICPC Programming Competition

- Nov 16, 2024

Let’s put it on our calendars!

QUESTIONS?

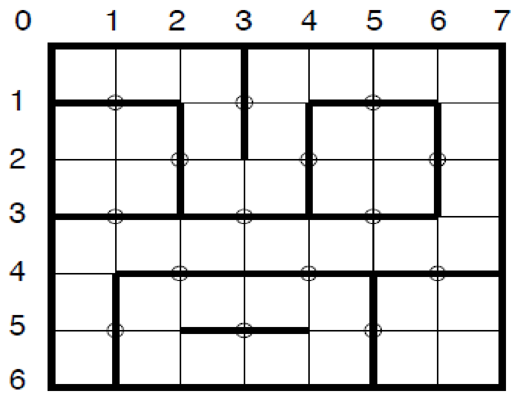
Fix the Pond

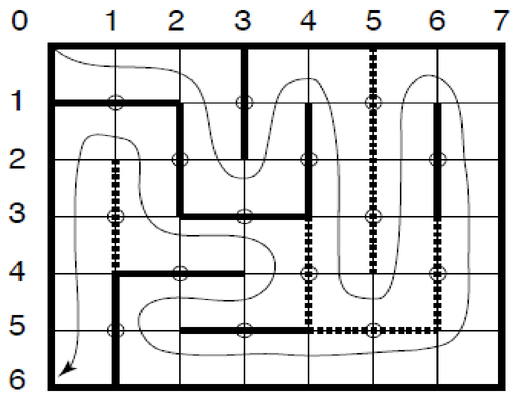
- https://vjudge.net/problem/UVA-12529

Problem F

Fix the Pond

A shrimp farm uses a rectangular pond built as a grid with $2N$ rows and $2N+1$ columns of square cells, for a given integer N. Each cell side is one meter long. The pond has exactly $(2N-1) \times N$ barriers of length two meters, used to temporarily isolate smaller sections inside the pond for breeding different kinds of shrimp. The barriers have their middle points fixed precisely at the integer coordinates (a, b), for all $0 < a < 2N$ and $0 < b < 2N+1$, where both a and b are odd, or both are even. Each barrier can be rotated around its middle point to change the pond configuration; however, by being rotated, a barrier switches between only two possible positions, always parallel to the pond sides, vertical or horizontal. The left part of the figure below shows a pond configuration, with $N = 3$.





At the end of every season the pond is closed for maintenance and cleaning. It must then be reconfigured so that a special machine can sweep the pond floor. The machine starts its work at the top left cell, and needs to pass through every cell exactly once, finishing in the bottom left cell. The right part of the figure shows one such reconfiguration, where six barriers were switched. For this example, though, four barrier switches would have been enough.

You must write a program that given a pond configuration, determines the minimum number of barrier switches needed to reconfigure the pond as specified above. There is always at least one possible way to reconfigure the pond as specified.

Input

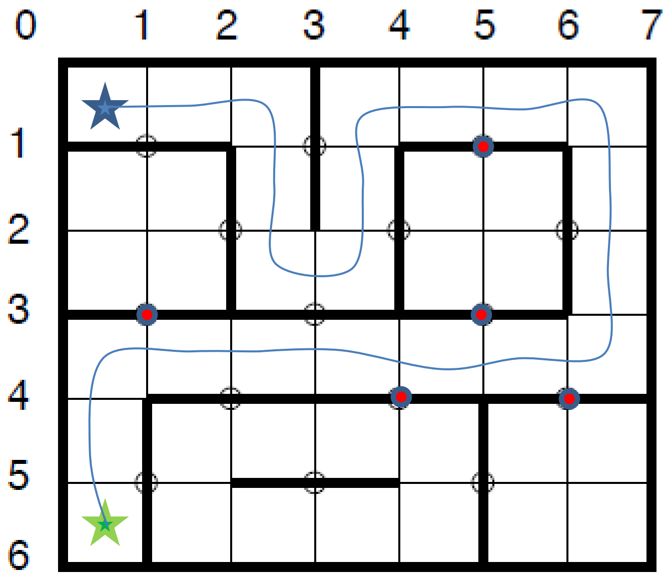
Each test case is described using several lines. The first line contains an integer N indicating that the pond has $2N$ rows and $2N+1$ columns ($1 \leq N \leq 300$). Each of the next $2N-1$ lines contains a string of N characters describing the orientation of the barriers. In the i-th line, the j-th character indicates the orientation of the barrier whose middle point is at coordinates $(i, 2j-1)$ if i is odd, or $(i, 2j)$ if i is even, for $i = 1, 2, \ldots, 2N-1$ and $j = 1, 2, \ldots, N$. The character is the uppercase letter "V" if the orientation is vertical, or the uppercase letter "H" if it is horizontal.

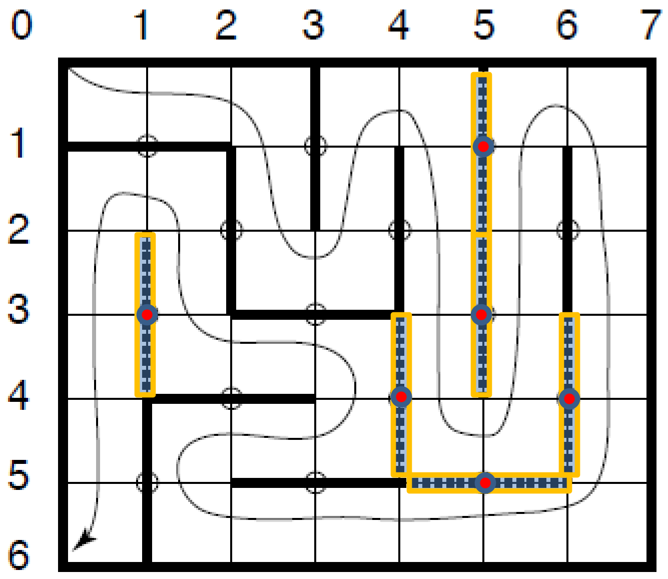
Output

For each test case output a line with an integer representing the minimum number of barrier switches needed to reconfigure the pond as specified.

Sample input	Output for the sample input
3 HVH VVV HHH HHH VHV 1 H 1 V	4 0 1

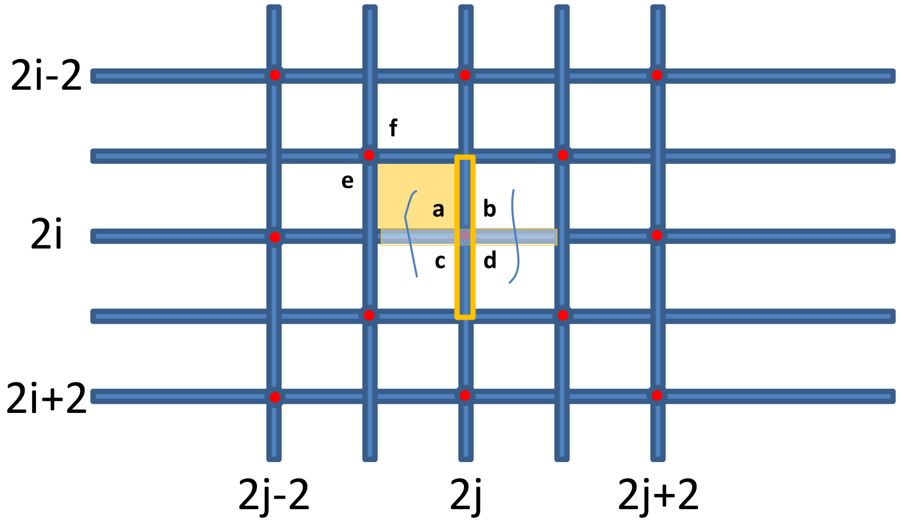
Uva 12529: Fix the Pond



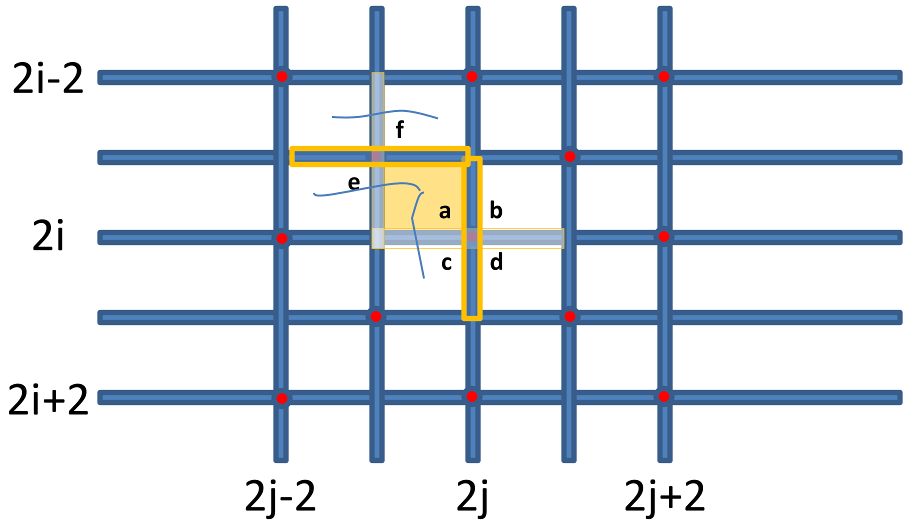


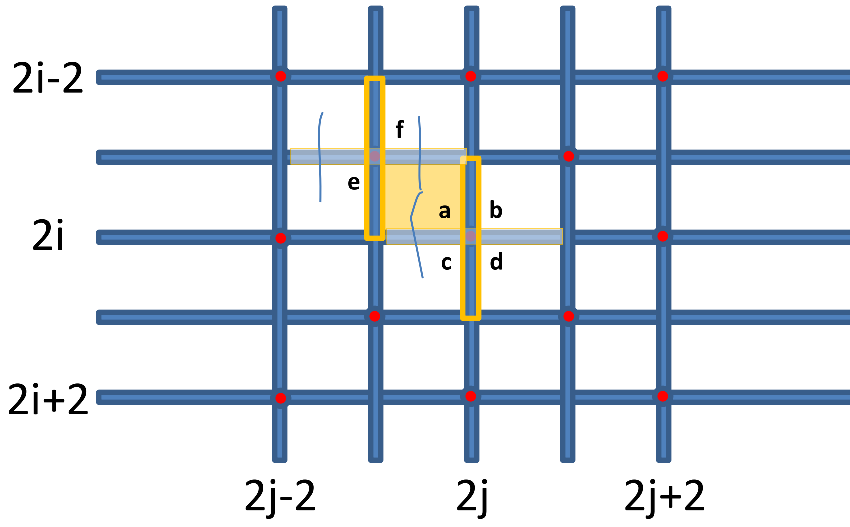
Neighbors of a pond cell?

Every cell has at most 2 neighbors at any given time. Start and end cell have at most 1 neighbor.

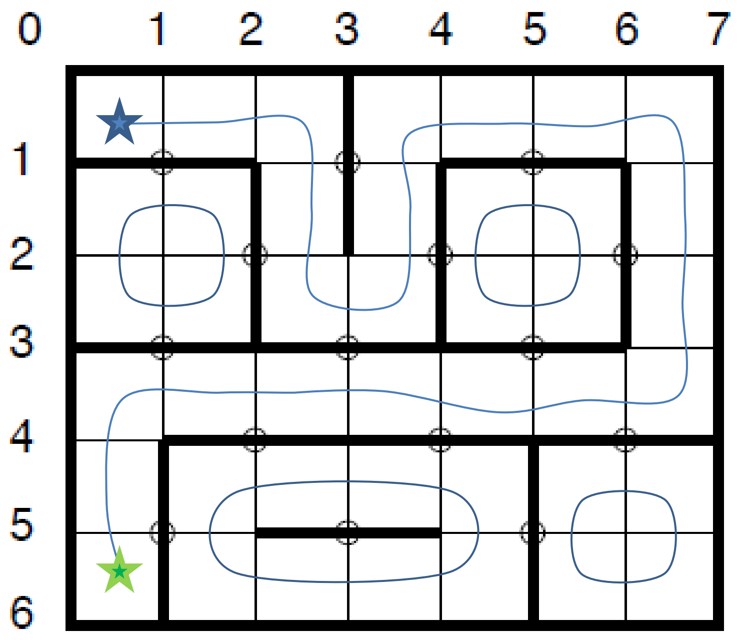


Effect of Flipping a Switch



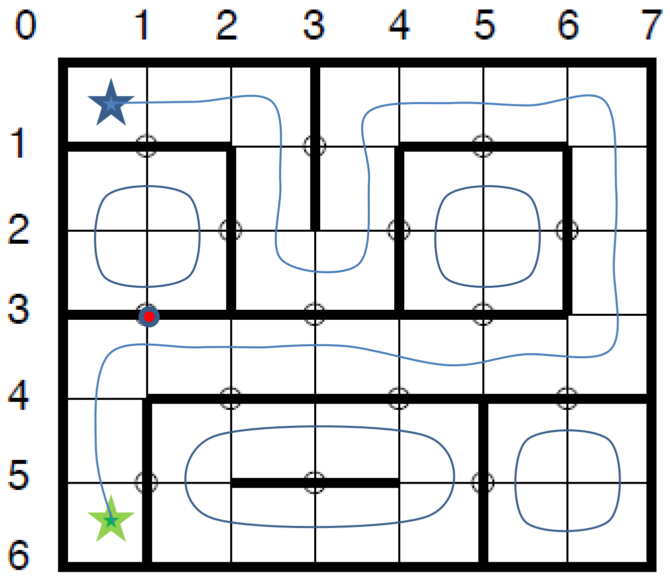


Observations



- Path P from top left (s) to bottom left (t)
- Path P always exists
  - Really?
- P may not visit all cells
  - Because of cycles
- Flipping switches can change path P

Flipping a Switch



Modeling as a Graph Problem

- Unable to visit every cell sometimes because graph may have disjoint cycles
- Path P from start (s) to end (t) exists. **Why**?
  - Each cell has two neighbors, except start & end
  - Graph = path P plus cycles (**Property $\boldsymbol{Q}$**)
- Flip a switch: path changes, but not above $\boldsymbol{Q}$
- What changes is # of cycles (c) in graph
- Every flip increases or decreases c by 1

Greedy Algorithm

- ~~Compute the undirected graph connecting cells~~
- ~~Find path P from top left (s) to bottom left (t)~~
- ~~Count # of cycles (c) in initial graph~~
- ~~For every switch that decreases # of cycles by 1 do~~
  - ~~Flip that switch~~
- ~~Report # of flips~~
  - ~~Such a switch always exists. Why?~~
  - ~~Switches that border P & an unreachable cell?~~
  - ~~Switches that border 2 unreachable cells (or inside one)?~~
- ~~# of flips = c. Why?~~
  - ~~After flipping c switches, # of cycles = 0~~

Time Complexity

- Run DFS & compute # connected components
  - O(**m+n**) = O(**n**)
  - Report # of connected components minus 1. Why?
- If we are asked to find exact switches to flip?
- Find switch to flip: walk along P & flip first switch bordering an unreachable cell
  - Total over all flips = O(**n**)