Διαγωνισμοί Αλγοριθμικού Προγραμματισμού για Φοιτητές

Αντικείμενο Διαγωνισμών

- Οι διαγωνιζόμενοι υλοποιούν προγράμματα τα οποία λύνουν δεδομένα προβλήματα.
- Η πλειονότητα των ερωτημάτων ανάγεται σε κάποιο πρόβλημα μαθηματικού ή λογικού χαρακτήρα.
- Απαραίτητες βασικές γνώσεις: συνδυαστικής, θεωρίας αριθμών, θεωρίας γραφημάτων, θεωρίας αλγορίθμων (π.χ., δυναμικός προγραμματισμός, εφαρμογή αλγορίθμων γραφημάτων), δομών δεδομένων.
- Διαδικασία επίλυσης ενός προβλήματος: α) σχεδίαση ενός αποδοτικού αλγόριθμου και β) υλοποίηση του σε κατάλληλη γλώσσα προγραμματισμού.
- Γλώσσες προγραμματισμού: C, C++, C#, Java, Python, κ.α.

Αξιολόγηση Λύσεων

- Η ορθότητα των λύσεων ελέγχεται αυτόματα από πρόγραμμα (judge) που εκτελεί μια σειρά από δοκιμαστικές περιπτώσεις.
- Ο κώδικάς θα πρέπει να περάσει όλες τις δοκιμές, δίνοντας τη σωστή απάντηση στο πρόβλημα.
- Στις περισσότερες περιπτώσεις, οι κριτές ελέγχουν και την απόδοση των προγραμμάτων. Π.χ., ο κώδικάς σας πρέπει να εκτελείται εντός των επιτρεπόμενων ορίων χρόνου, μνήμης (ή κάποιας άλλης παραμέτρου).

Γιατί να συμμετάσχω;

Οι διαγωνισμοί αλγοριθμικού προγραμματισμού προσφέρουν την ευκαιρία στους φοιτητές να αναπτύξουν και να αναδείξουν:

- Ικανότητες επίλυσης προβλημάτων.
- Σχεδίαση, ανάλυση και εφαρμογή αλγορίθμων, δίνοντας έμφαση στην ορθότητα και αποδοτικότητά τους.
- Δεξιότητες στον προγραμματισμό, τον έλεγχο και τον εντοπισμό σφαλμάτων.

Διαγωνισμοί

- International Collegiate Programming Contest: https://icpc.global/
- IEEEXtreme: https://ieeextreme.org/
- Google competitions: https://codingcompetitions.withgoogle.com/
- Top Coder: https://www.topcoder.com/talent/compete/
- Kattis: https://open.kattis.com/
- Code Forces: https://codeforces.com/
- UVa Online Judge: https://onlinejudge.org/
- Code Chef: https://www.codechef.com/
- Project Euler : https://projecteuler.net/
- USA Computing Olympiad Training Program: https://train.usaco.org/usacogate

Χρήσιμο Υλικό

- Online book: Principles of Algorithmic Problem Solving
- Competitive programming website: https://cp-algorithms.com/

acm International Collegiate Programming Contest

icpc 2018 World Finals

icpc.foundation

Problems Beijing

Problem J
Uncrossed Knight’s Tour
Time limit: 2 seconds

A well-known puzzle is to “tour” all the squares of an 8×8 chessboard using a knight, which is a piece that can move only by jumping one square in one direction and two squares in an orthogonal direction. The knight must visit every square of the chessboard, without repeats, and then return to its starting square. There are many ways to do this, and the chessboard size is manageable, so it is a reasonable puzzle for a human to solve.

However, you have access to a computer, and some coding skills! So, we will give you a harder version of this problem on a rectangular $m \times n$ chessboard with an additional constraint: the knight may never cross its own path. If you imagine its path consisting of straight line segments connecting the centers of squares it jumps between, these segments must form a simple polygon; that is, no two segments intersect or touch, except that consecutive segments touch at their common end point. This constraint makes it impossible to visit every square, so instead you must maximize the number of squares the knight visits. We keep the constraint that the knight must return to its starting square. Figure J.1 shows an optimal solution for the first sample input, a 6×6 board.

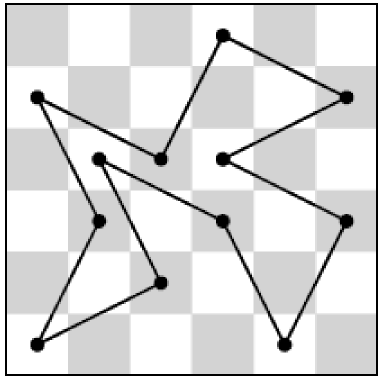


Figure J.1: An optimal solution for a 6×6 board.

icpc.foundation

the 45th annual World Finals Dhaka

icpc international collegiate programming contest

Problem D
Guardians of the Gallery
Time limit: 5 seconds

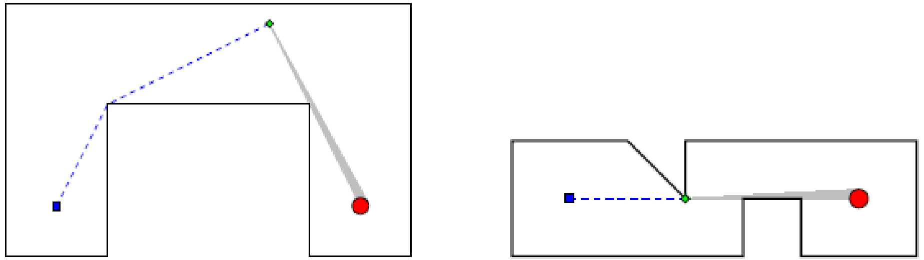
Your local art gallery is about to host an exciting new exhibition of sculptures by world-renowned artists, and the gallery expects to attract thousands of visitors. Unfortunately, the exhibition might also attract the wrong kind of visitors, namely burglars who intend to steal the works of art. In the past, the gallery directors did not worry much about this problem, since their permanent collection is, to be honest, not really worth stealing.

The gallery consists of rooms, and each sculpture in the new exhibition will be placed in a different room. Each room has a security guard and an alarm to monitor the artwork. When an alarm sounds, the guard will run (without leaving the room) from their post to a position where they can see the sculpture directly. This is to check whether the sculpture has in fact been stolen, or whether this is yet another false alarm.

To figure out where to best station the security guard, the gallery directors would like to know how long it takes for the guard to see a given sculpture. They hope that you can help!

Every room is on a single floor, and the layout of the walls can be approximated by a simple polygon. The locations of the guard and the sculpture are distinct points strictly inside the polygon. The sculpture is circular, with a negligibly small (but positive) radius. To verify that the sculpture is still present, the guard needs to be able to see at least half of it.

Figure D.1 illustrates two examples. In each case, the guard starts at the blue square on the left, and the sculpture is located at the red circle on the right. The dotted blue line shows the optimal path for the guard to move. Once the guard reaches the location marked by the green diamond, half of the sculpture can be seen.



(a) Sample Input 1

(b) Sample Input 2

Figure D.1: Illustration of sample inputs.