Анализ на задача "Огледална табла"

CodeIT, 2014-2015, Round 6

Първото, което трябва да забележим е, че за да преместим всички черни пулове в една дъска трябва да няма два огледални черни пула.

Ако това е изпълнено, то броят ходове, необходими за да преместим всички черни пулове от една дъска в друга, е равен на броя черни пулове върху тази дъска. Затова ако преброим черните пулове върху двете дъски можем просто да изберем тази, върху която има по-малко черни пулове.

За илюстрация ще използвам фрагмент от авторовото решение (C++):

1. // големината на дъските
2. int n;
3. // mat[ред][колона] - двумерен масив с пуловете
4. // първата дъска е: ред = 0 ... n-1
5. // втората дъска е: ред = n ... 2n-1
6. // 0 = бял пул, 1 = черен пул
7. int mat[2 \* MAX\_N][MAX\_N];
8. ...
9. void solve ()
10. {
11. // броя черни пулове върху всяка дъска
12. int countBlack[2] = {0, 0};
13. // обхождаме всяка клетка на първата дъска
14. for (int row = 0; row < n; row++)
15. {
16. for (int col = 0; col < n; col++)
17. {
18. // намираме огледалният ред
19. int invRow = 2\*n - row - 1;
20. // проверяваме дали има два черни пула на огледални позиции
21. // ако е така принтираме резултата -1 и прекъсваме функцията
22. if (mat[row][col] && mat[invRow][col])
23. {
24. printf("-1\n");
25. return;
26. }
27. // обновяваме броя черни пулове във всяка дъска
28. // в масива mat: 0 = бял пул, 1 = черен пул
29. countBlack[0] += mat[row][col];
30. countBlack[1] += mat[invRow][col];
31. }
32. }
33. // ако всичко е минало наред при обхождането
    * 1. // и сме преброили черните пулове върху всяка дъска
34. // принтираме по-малкият брой пулове от дветата
35. printf("%d\n", std::min(countBlack[0], countBlack[1]));
36. }
37. Автор: Никола Стоянов